

IMPLEMENTATION OF AN EXCITATION ANALYZER FOR A SPEECH ANALYSIS/SYNTHESIS SYSTEM

**A Thesis Submitted
In Partial Fulfilment of the Requirements
for the Degree of
MASTER OF TECHNOLOGY**

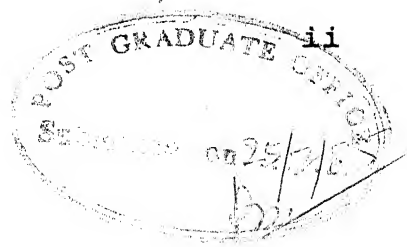
**By
CAPT. CVSS SATISH**

**to the
DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR
MARCH, 1985**

10 JUN 1985

87599

EE-1985-M-SAT-IMP



CERTIFICATE

This is to certify that the thesis entitled
'IMPLEMENTATION OF AN EXCITATION ANALYZER FOR A SPEECH
ANALYSIS-SYNTHESIS SYSTEM' submitted by Capt. CVSS Satish,
Roll No. 8310448, for the partial fulfilment of the require-
ments for the award of M.Tech. degree, has been carried out
under our supervision. To the best of our knowledge, it
has not been submitted elsewhere for an award of a degree.

P.K. Chatterjee

(P.K. Chatterjee)
Professor
Dept. of Electrical Engg.
Indian Institute of Technology
Kanpur

K.R. Srivathsan

(K.R. Srivathsan)
Assistant Professor)
Dept. of Electrical Engg.
Indian Institute of Technology
Kanpur

13/05/85

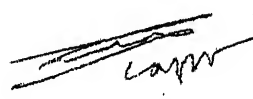
ACKNOWLEDGEMENTS

I am truly grateful to Dr. K.R. Srivathsan and Dr. P.K. Chatterjee for their guidance in the completion of this thesis.

Dr. Srivathsan, in particular, has taken a lot of trouble in ironing out the hardware kinks whenever they occurred.

Thanks are due to all the occupants of the 3rd Floor, ACES for their excellent company and the congenial atmosphere prevalent. RES Prem Malhotra and GNMS Sudhakar have rendered a lot of help in making my passage smooth. Brain storming sessions with Nandi and Mukund were invaluable.

Mr. R.N. Srivastava must get all the credit for his typing effort and the trouble he has taken to decipher my scrawl.



CVSS Satish
Captain

ABSTRACT

Various schemes for digital processing of speech signals is reviewed. Those schemes employing parametric analysis of speech data, viz., the vocal tract and excitation functions, provide for efficient coding and data rate compression.

An LPC analysis/synthesis system is taken up in detail and its hardware implementation is suggested. This scheme consists of two parts: — a pitch period estimator and an LPC analyzer.

An excitation analyser needed to obtain the pitch period estimates of a speech signal is implemented. This implementation uses a special purpose, signal processing microcomputer, the Intel 2920. Signal processing functions such as digital Lerner filters, peak detectors and run down circuits are implemented on this processor. The pitch estimates and voiced/unvoiced decisions are made using the Parallel Processing method of Gold and Rabiner. An 8085 microprocessor is used for the final computation of the pitch period from parallel estimates in an interrupt service routine. The estimator does provide fairly accurate results.

CONTENTS

CHAPTER 1 :	SPEECH PROCESSING	1
1.1	Introduction	1
1.2	Speech Processing	1
1.3	Rate Compression of Speech	2
1.4	Speech Processing Techniques	2
1.4.1	Waveform Coding	3
1.4.2	Parametric Coding	3
1.5	Model of Speech Waveforms	5
1.5.1	The Speech Mechanism	5
1.6	Short Time Analysis and Synthesis of Speech	6
1.6.1	Requirement of the Short Time Basis	6
1.6.2	Duration of the "Short Time"	7
1.7	Thesis Outline	7
CHAPTER 2 :	REVIEW OF SPEECH ANALYSIS SYNTHESIS SYSTEMS	9
2.1	Speech Analysis-Synthesis Systems	9
2.2	Analysis Techniques	9
2.2.1	Short Time Autocorrelation Analysis	10
2.2.2	Short Time Fourier Analysis	11
2.2.3	Homomorphic Analysis	14
2.2.4	Formant Analysis	16
2.2.5	Analysis by Linear Prediction	18
2.2.5.1	Autocorrelation Method	22
2.3	Durbin's Recursive Solution	23
2.4	Synthesis Methods	24
2.4.1	Terminal Analog Synthesizer	25
2.4.2	Acoustic Tube Analog Synthesizer	25
2.5	Vocoders	27
2.5.1	Channel Vocoder	28
2.5.2	Homomorphic Vocoder	28
2.5.3	Vocoders Based on Formant Analysis	28
2.5.4	LPC Vocoder	31
2.6	Summary	33

CHAPTER 3 :	EXCITATION ANALYZERS	34
3.1	Pitch Period Estimation	34
3.2	Selection for Implementation	39
3.3	The Parallel Processing Algorithm	39
3.4	The implemented Modification	
CHAPTER 4 :	IMPLEMENTATION OF THE PARALLEL PROCESSING PITCH PERIOD ESTIMATOR	45
4.1	Considerations for the Design	45
4.2	The Initial Attempt	45
4.3	The Final Implementation	47
4.3.1	The Hardware	47
4.3.1.1	Use of 2920 Signal Processor	48
4.3.1.2	The Pitch Period Estimators	50
4.3.1.3	The Interrupt Set-up	52
4.3.2	The Software	52
4.3.2.1	The Signal Processor Software	53
4.3.2.2	Final Computation Software	56
CHAPTER 5 :	SUGGESTIONS FOR FURTHER WORK	60
5.1	Improvements for Present Implementation	60
5.2	Suggested Scheme for the Realisation of LPC Vocoder	62
5.2.1	Analyzer Implementation	62
5.2.2	Synthesizer Implementation	64
5.3	Conclusion	65
REFERENCES		67
APPENDIX A :	CIRCUIT DIAGRAMS	
APPENDIX B :	PROGRAM IMPLEMENTED ON THE 2920	
APPENDIX C :	FINAL COMPUTATION PROGRAM	
APPENDIX D :	2920 SIGNAL PROCESSOR NOTES	

CHAPTER 1

SPEECH PROCESSING

1.1 INTRODUCTION

Digital processing of speech has been an active subject of study for several decades. It has acquired added importance with digital transmission of speech and secure communications. The attraction has resulted from the wide variety of applications from communications to automatic reading machines [1]. In this thesis various speech analysis systems have been discussed briefly. In detail, is available the implementation of a pitch period estimator, which forms an important part of any speech analysis-synthesis system.

1.2 SPEECH PROCESSING

Speech processing problems fall broadly into three classes depending on application [3, 4].

a) Speech analysis - for such applications as

- speech recognition
- speaker identification
- speaker verification

b) speech synthesis - for application in

- automatic reading machines
- data retrieval, verbally, from a computer as when a data base is to be interrogated from an ordinary telephone.

c) Analysis followed by synthesis - for voice communication uses with the aim of

- secure voice transmission
- data rate compression of speech.

It is this third class of problems that this work addresses itself to.

1.3 RATE COMPRESSION OF SPEECH

By simply sampling and digitising speech we have a rate of the order of 64 K bits/sec. However, through the use of speech analysis followed by appropriate coding at the transmission end and resynthesis at the receiver, this can be reduced by a factor between 10 and 50. The reduction factor depends on the type of system used and the speech quality [3] desired. Intelligible speech can be communicated at as low as 2.5 K bits/sec. For accommodating nuances of speech such as inflection, tone and auditory hints to the speakers identity, which are desirable in human communications, higher data rates must be used to accommodate this information.

1.4 SPEECH PROCESSING TECHNIQUES

The techniques available for speech processing are broadly divided into two classes:

- (a) Waveform coding methods
- (b) Methods based on the structure of speech.

1.4.1 Waveform Coding

In waveform coding methods [2] the only assumption made is that the signal is bandlimited. This class can in general be applied to any bandlimited time function. These are the PCM, DPCM and DM techniques. If adaptive quantizers are incorporated in these, we have the APCM, ADPCM and ADM with better performance at a higher cost. Mention may also be made of the LDM (linear delta modulation), CDM (continuous DM) and the DCDM (digitally controlled DM). In all, these techniques provide higher SNR at data rates of 30 to 50 K bits/sec. The perceptual speech quality is dependent on the quantisation technique and the number of bits per sample. A study [2, 4] shows that objective ratings, based on SNR considerations and subjective ratings based on listener preferences can vary, as shown in Figure 1.1.

Here it is evident that transmission rates between 24 to 56 K bits/sec are needed for a sample rate of 8 KHz, to obtain a reasonable quality of speech.

1.4.2 Parametric Coding

This class of techniques is tailored for the speech waveform in the sense that they capitalise on its structure, when represented by a model consisting of a slowly varying linear system excited by an appropriate excitation signal. To mention a few of the methods [1, 3, 4] in this class we have the Linear Predictive Coding (LPC), Homomorphic filtering and Formant analysis. These methods are attractive

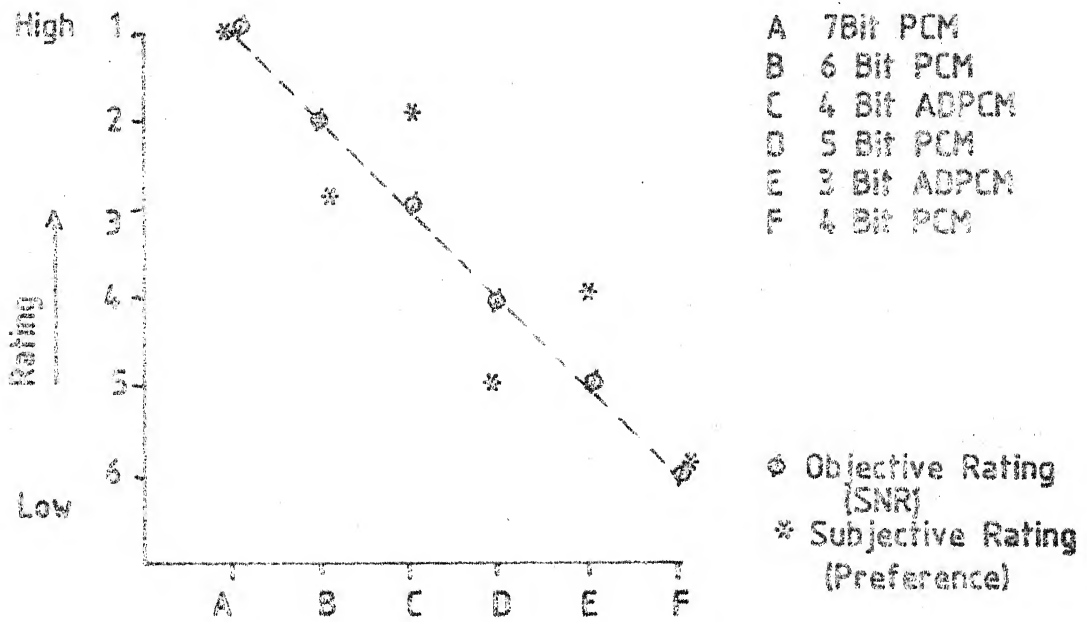


Fig 1.1 Comparison of Objective and Subjective Performance of ADPCM and log PCM

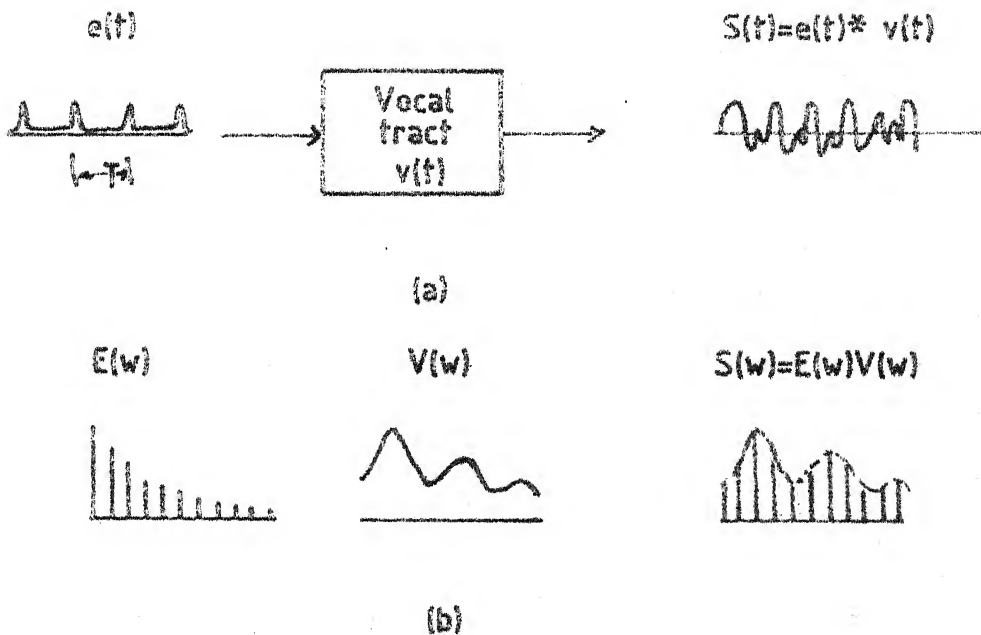


Fig 1.2 Model of speech production

- (a) Time domain characterisation
- (b) Frequency domain

since they offer lower bit rates by transmitting only the speech parameters rather than the actual waveform itself. These are discussed in the next chapter.

1.5 MODEL OF SPEECH WAVEFORMS

The second class of techniques for speech analysis and synthesis can be viewed in terms of a model of the speech waveform as the response of a slowly time varying system to either a periodic or noiselike excitation.

1.5.1 The Speech Mechanism

The speech mechanism essentially consists of:

(a) The vocal tract which is an acoustic tube with non uniform cross sectional area, ranging from the vocal chord constriction at the mouth of the trachea to the lip at the other end. During speech the vocal tract is deformed in cross sectional area by movement of the articulators namely, the lips, jaw, tongue and the velum (soft palate).

(b) The excitation source or, the source of energy for the speech production lies in the thoracic and abdominal musculatures. In the case of voiced speech sounds, the excitation corresponds to a quasi periodic pulse train, representing air flow through the vocal chords as they vibrate. The fricative sounds are generated by forcing air through a constriction in the vocal chords, creating turbulence which in turn produces a source of noise to excite the vocal tract [1].

As suggested in the preceding discussion, the speech waveform can be modelled as the response of a linear time varying system, the vocal tract, with the appropriate excitation.

Figure 1.2(a) shows that for a fixed vocal tract, the output of the system is the convolution of vocal tract and excitation impulse responses. The vocal tract changes for different sounds, and since this change is a slow one, the output can be approximated as a convolution on a short time basis.

1.6 SHORT TIME ANALYSIS AND SYNTHESIS OF SPEECH

1.6.1 Requirement of the Short Time Basis

Figure 1.2(b) shows the model in the frequency domain where the speech output is a composite spectrum of the product of a line spectrum (Fourier transform (FT) of the excitation for a voiced signal) and the FT of the vocal tract impulse response.

To generate different sounds, the vocal tract shape and hence the envelope of the output spectrum changes. Similarly, as the excitation period changes for voiced sounds, the spacing of the pitch harmonics will change.

Since both the components of the output product change, albeit slowly, with time, a short time analysis is desirable and meaningful.

Identically the short time basis is implied during synthesis for which purpose, periodic information of the

excitation and vocal tract parameters are adequate.

1.6.2 Duration of the "Short Time"

A natural question to follow at this stage is how short or long can the "short time" be. Many works in the area suggest that the time interval lie between 10 to 40 msec. Instinctively more accurate representation of analysed speech is better obtained at smaller intervals. However, subjective listener tests [1, 5] indicate a wide variance depending upon the analysis-synthesis methods used. Once again desired speech quality can determine the duration of the "short time", along with the restrictions imposed by the method of analysis-synthesis in consideration.

1.7 THESIS OUTLINE

The region of study is the Parametric coding methods. In this thesis an attempt to study the hardware implementation of one type of Parametric coding technique is made.

Chapter 2 introduces and **reviews** the various types of analysis-synthesis methods available and their implementation in vocoders.

Chapter 3 mentions available schemes for the extraction of the excitation parameter, or pitch period, at block diagram level. The Parallel Processing Algorithm of Gold and Rabiner [6] is discussed along with the modification chosen for implementation.

Chapter 4 covers in detail the hardware and software for the implementation of the Parallel processing pitch

period estimation algorithm.

Chapter 5 contains suggestions for improvement of the implemented design and also a scheme for the realisation of a complete LPC vocoder.

Appendix A contains the hardware details.

Appendix B contains printouts of software programmed on to the 2920 Signal Processing Chip.

Appendix C has the printout of the 8085 program used for final computation of the pitch period.

Appendix D contains notes on the use of 2920 Signal Processor and Software Application Package.

CHAPTER 2

REVIEW OF SPEECH ANALYSIS SYNTHESIS SYSTEMS

2.1 SPEECH ANALYSIS-SYNTHESIS SYSTEMS

To capitalise on the model of speech presented in Chapter 1, most analysis-synthesis methods attempt, in one way or another, to deconvolve the speech signal to achieve separation of the vocal tract characteristics from the excitation function. Such systems lead to significant data rate compression for speech storage or transmission at the cost of quality. This results from the fact that the deconvolution cannot be precisely implemented [3]. Several methods of analysis are available for implementation in speech processing systems. The synthesis can then be performed in one of the synthesizers available.

2.2 ANALYSIS TECHNIQUES

A variety of analysis methods are possible based on the output of a slowly varying linear system. Those of interest are

- a) Short time autocorrelation analysis [11]
- b) Short time Fourier analysis [3, 4, 11]
- c) Homomorphic analysis [3, 4, 9, 11]
- d) Formant analysis [1, 14]
- e) Analysis by linear prediction [5, 12, 15]

2.2.1 Short Time Autocorrelation Analysis

This is a time domain analysis method wherein the utility of the autocorrelation function in displaying structure in any waveform is employed.

The autocorrelation function of a discrete time signal $x(n)$ is defined as

$$\phi(m) = \lim_{N \rightarrow \infty} \frac{1}{2N + 1} \sum_{n=-N}^N x(n) x(n + m) \quad (2.1)$$

If the signal is periodic with period p i.e. $x(n + p) = x(n)$ for all n , then it is easily shown that

$$\phi(m) = \phi(m + p) \quad (2.2)$$

Thus periodicity in the autocorrelation function indicates periodicity in the signal. A lack of predictable structure in a signal is indicated by an autocorrelation function sharply peaked at $m = 0$ and falling off rapidly as m increases.

Using the notion of short time analysis to operate on short segments of the speech signal, the short time autocorrelation function can be defined as

$$\phi_1(m) = \frac{1}{N} \sum_{n=0}^{N-1} x_1(n) x_1(n + m), \quad 0 \leq m \leq M_0 - 1 \quad (2.3)$$

where N = number of samples of a segment

l = beginning of the segment

M_0 = maximum lag of interest and is $>P$ if periodicity is to be observed.

If $N' = N$, then data from outside the segment $1 \leq n \leq N + 1 - 1$ is used in computation.

If $N' = N - m$, then data from that interval only is required and the segment is often weighted by a "window" function that smoothly tapers the ends of the segment to zero. Either choice is satisfactory to detect periodicity in the speech.

The direct computation of $\phi_1(m)$ for $0 \leq m \leq M_0 - 1$ requires computational effort proportional to $M_0 \times N$, which can be a significant overhead.

The estimate of the autocorrelation is generally based upon 20 to 40 millisecond segments, making allowance that the window must be long enough to encompass at least two periods of the speech signal.

2.2.2 Short Time Fourier Analysis

This spectral analysis method is classic to the approach of obtaining the vocal tract transfer function. Speech is a quasi-stationary process and may be considered stationary for adequately short segments. Therefore the FT of a short segment of speech provides a good spectral representation of it during that interval.

Figure 2.1 shows a simple way of implementing a short time spectral analyzer. The implementation involves using a bank of bandpass filters. Choosing the filter passbands to cover the speech band, the outputs can be thought of as a Fourier representation of the speech

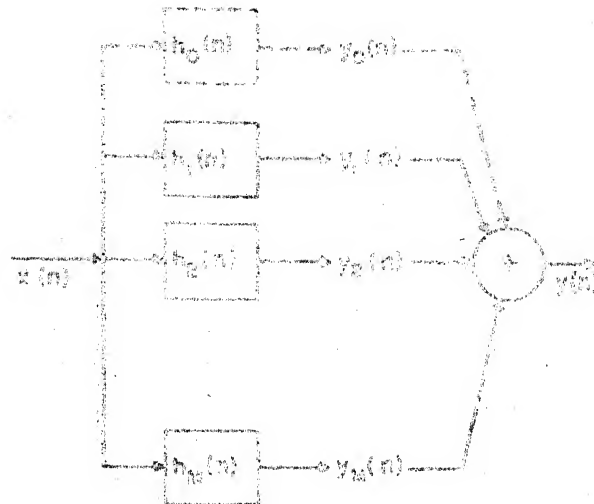


Fig. 2.1 A bank of bandpass filters.

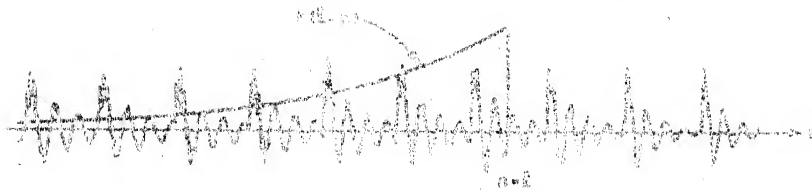


Fig. 2.2 Illustration of computation of the short-time Fourier transform.

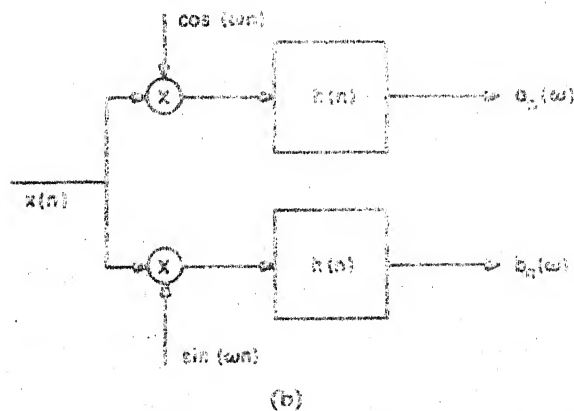
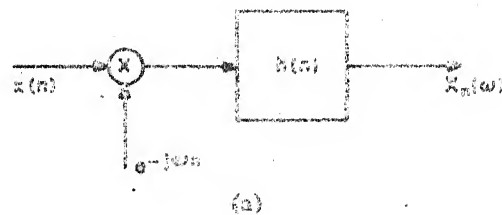


Fig. 2.3 Short-time Fourier analysis and synthesis for one channel centered at ω .

signal. With carefully designed filters, the sum of all outputs will be a good approximation to the original speech signal.

The discrete short time spectrum of $x(n)$ is defined as

$$X_1(w) = \sum_{n=-\infty}^{\infty} x(n) h(1-n) e^{-jwn} \quad (2.4a)$$

$$= |X_1(w)| e^{j\theta_1(w)} \quad (2.4b)$$

$$= a_1(w) - j b_1(w) \quad (2.4c)$$

Two basic interpretations can be made from equation (2.4a):

(a) $X_1(w)$ may be viewed as the FT of the sequence $x(n)$ weighted by a window $h(1-n)$ as in Figure 2.2.

(b) The second interpretation follows the assumption that $h(x)$ be the impulse response of a low pass digital filter. Assuming that it is desired to evaluate the short time transform at frequency w , from Figure 2.3(a) $X_n(w)$ is the output of a low-pass filter with input $x(n)e^{-jwn}$. The depiction in Figure 2.3(b) avoids complex arithmetic and the output parameters obtained i.e. $a_n(w)$ and $b_n(w)$, are the real and imaginary parts of the spectrum respectively.

The choice of bandwidth of the bandpass filters of Figure 2.2 is discussed in Section 2.5.1 where the implementation of the short time spectral analysis is realised in the channel vocoder.

The short time spectral analysis may be performed using a FFT algorithm. When implemented on a computer, the FFT method is generally superior to the bank of filters model.

2.2.3 Homomorphic Analysis

Homomorphic filtering is a class of non linear signal processing techniques that is based on a generalisation of the principle of superposition that defines linear systems [3]. It is a tool used to separate signals that have been non additively combined. Hence it serves to deconvolve the vocal tract and excitation functions.

The basic operations of such an analyzer are depicted in Figure 2.4(a). The signal at A is taken as a discrete convolution of the excitation and vocal tract impulse response. B is the result obtained by using a FFT and is the product of the FTs of vocal tract and impulse response. C is logarithm of the magnitude of the FT and is the sum of the logarithms of the excitation and vocal tract responses. Since the inverse DFT performed is linear, the result at D (called the cepstrum of input at A) is an additive combination of cepstra of the excitation and vocal tract components. Thus we have approximately transformed convolution to addition.

The cepstrum (D) serves as an excellent basis for estimating the fundamental period of voiced speech and for determining whether a particular speech segment is voiced or unvoiced [13].

The vocal tract transfer function, or the spectrum envelope, can be obtained by removing the rapidly varying

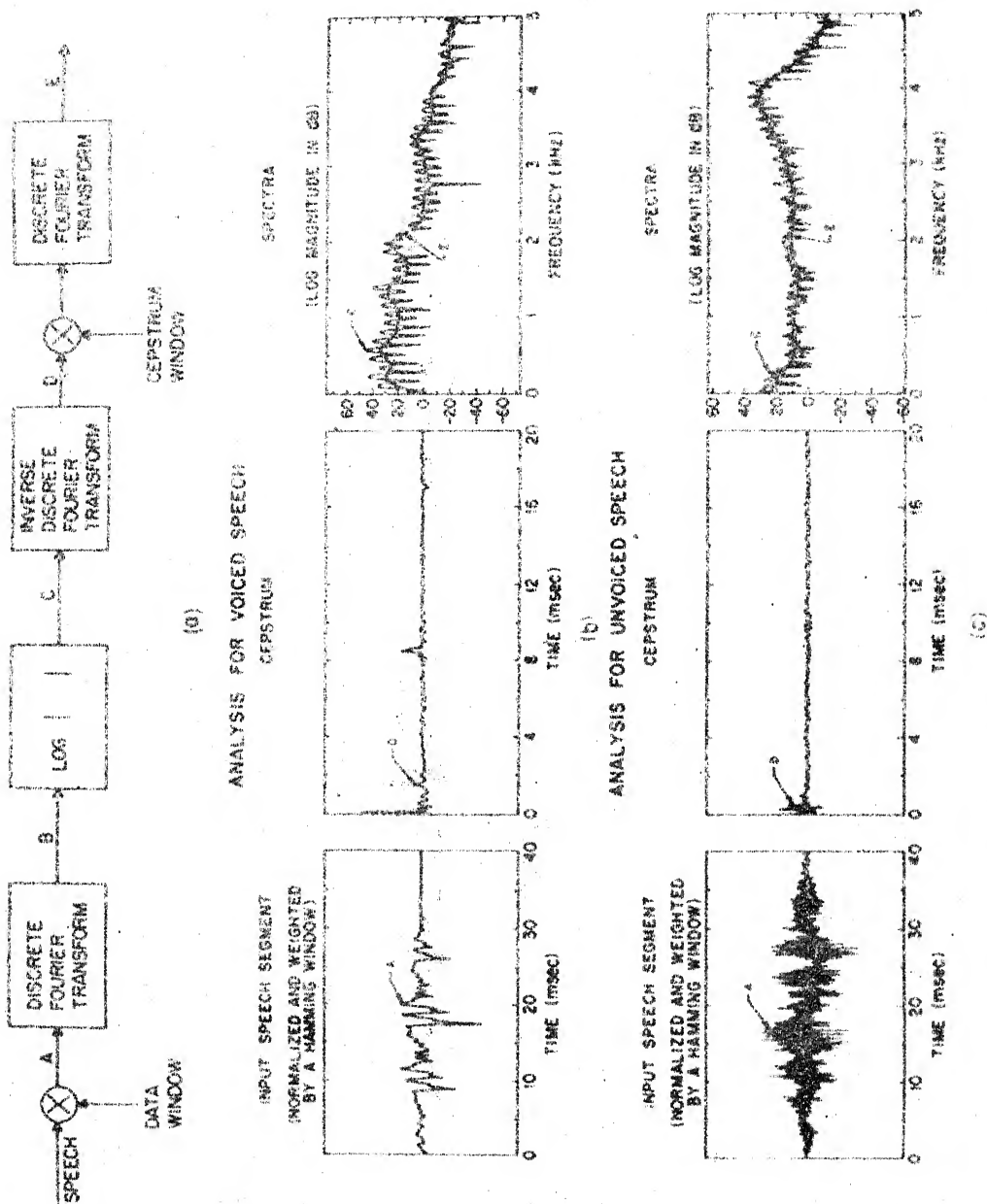


Fig. 2.4 Homomorphic processing of speech. (a) Basic operations. (b) Analysis for voiced speech. (c) Analysis for unvoiced speech.

components of the log magnitude spectrum by linear filtering. One method is to multiply the cepstrum at D by a window that only passes the short time components and then computing the DFT resulting in E (Figure 2.4(b) and (c) for voiced and unvoiced speech segments).

2.2.4 Formant Analysis

This method devolves around a time-frequency-intensity display of the short time spectrum of speech known as a "speech spectrogram". The vocal tract has resonances for voiced speech. The formant frequency is the frequency of the maximum of a gross concentration of energy in the spectrum of a speech sound. Fig. 2.4(d) is one such spectrogram. A wide band spectrogram is preferred over a narrow band one [3, 7] since time resolution is relatively high and, in fact, the individual periods of the time waveform are evident. During voiced intervals the vocal tract resonances appear clearly as dark bands in the spectrogram.

Formant analysis consists of a system which accepts speech as an input and yields output voltages whose magnitudes, as functions of time, represent the frequencies of the formants. During the silent and unvoiced intervals of speech utterances the output voltages should be extrapolated continuously [14].

Such formant extraction is put to use in the terminal analog synthesiser. It also provides an interesting tool for clinical and therapeutical speech work. A large number of schemes have therefore been devised for formant

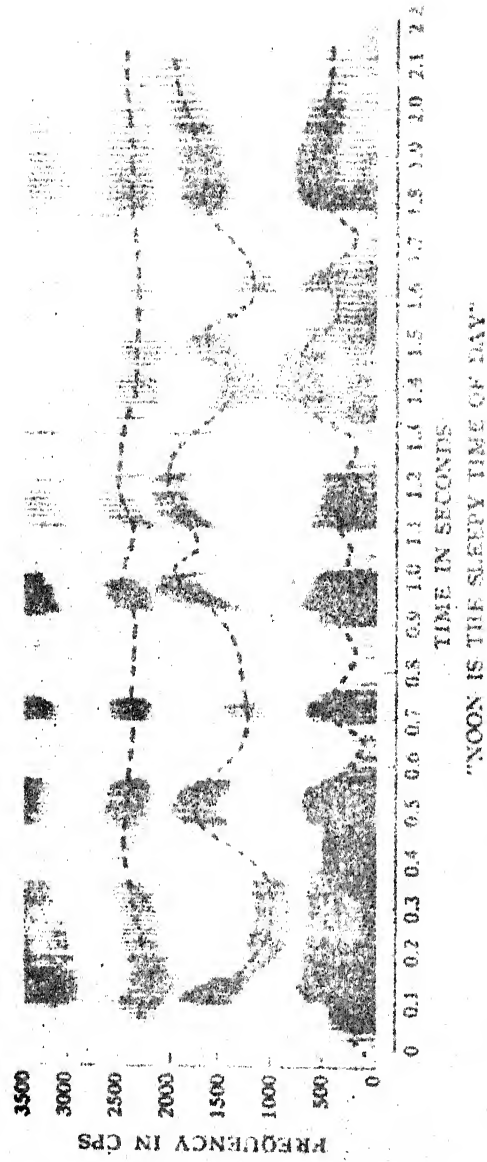


Fig 2.4(d) A Spectrogram of an utterance

frequency extraction, including by linear predictive analysis [5].

2.2.5 Analysis by Linear Prediction

An alternative to the above methods is an approach based on estimating parameters of a vocal tract model. One such representation is in terms of a general rational transfer function of the form

$$H(Z) = G \frac{1 + \sum_{l=1}^q b_l Z^{-l}}{1 - \sum_{k=1}^p a_k Z^{-k}} \quad (2.5)$$

A speech segment is sufficiently complex that it cannot be expected to match exactly the above model of equation (2.5). Much less, will be the matching to simplified all pole or all zero models. However, since a vocal tract transfer function is primarily characterised by resonances, it is fairly reasonable that an all pole model will preserve the important characteristics of the vocal tract transfer function. Such an all pole, i.e. Autoregressive, modelling technique is called "Linear Prediction" [3, 5, 11, 12, 15].

A simplified modelling on this basis is illustrated in Figure 2.5.

Consider that the time varying filter of Figure 2.5 has a steady state system function of the form

$$H(Z) = \frac{X(Z)}{U(Z)} = \frac{G}{1 - \sum_{k=1}^p a_k Z^{-k}} \quad (2.6)$$

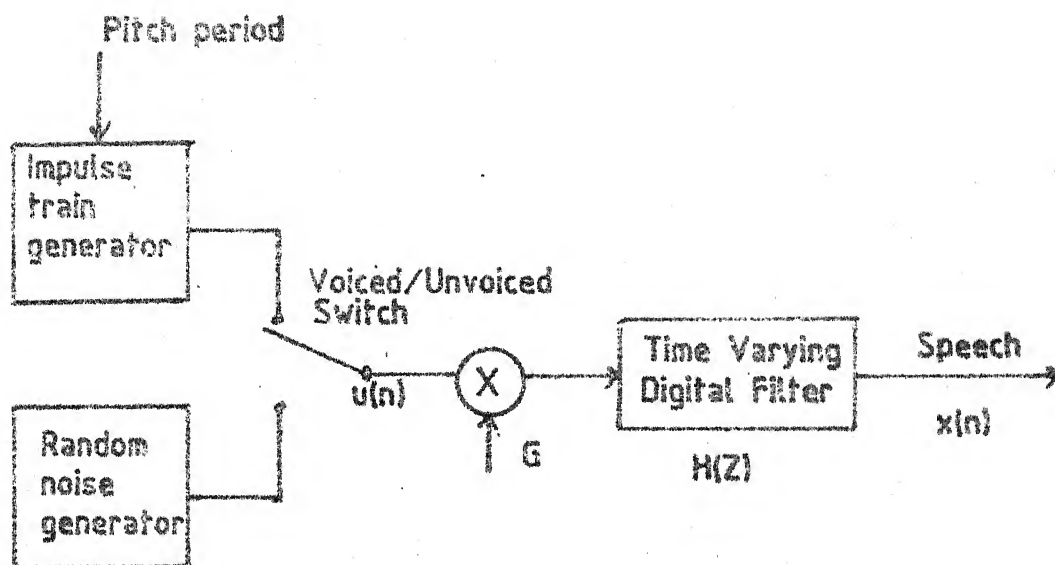


Fig 2.5 Block Diagram of a simplified model for speech production

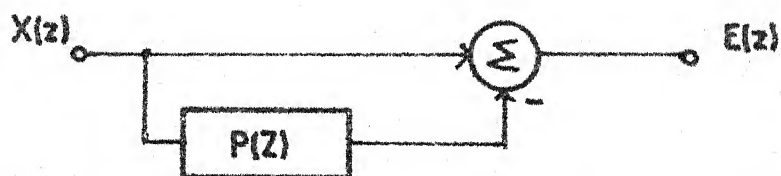


Fig 2.6 Linear Prediction model

$$\text{where } X(Z) = \sum_{n=-\infty}^{\infty} x_n Z^{-n} \quad (2.7)$$

The speech samples $x(n)$ are related to the excitation $u(n)$ by the difference equation

$$x(n) = \sum_{k=1}^p a_k x(n-k) + G u(n) \quad (2.8)$$

where G is the gain factor.

Define a linear predictor system (as in Figure 2.6) with predictor coefficients α_k , as a system whose output is

$$\hat{x}(n) = \sum_{k=1}^p \alpha_k x(n-k) \quad (2.9)$$

The system function of the p th order linear predictor is the polynomial

$$P(Z) = \sum_{k=1}^p \alpha_k Z^{-k} \quad (2.10)$$

The prediction error, $e(n)$ is defined as

$$e(n) = x(n) - \hat{x}(n) = x(n) - \sum_{k=1}^p \alpha_k x(n-k) \quad (2.11)$$

From equation (2.11) it is seen that the prediction error sequence is the output of a system whose transfer function is

$$A(z) = 1 - \sum_{k=1}^p \alpha_k z^{-k} \quad (2.12)$$

$$\text{or } A(z) = 1 - P(z) \quad (2.13)$$

Thus the prediction error filter, $A(z)$ will be an inverse filter for the system, $H(z)$ of equation (2.6), i.e.

$$H(z) = \frac{G}{A(z)} \quad (2.14)$$

The basic problem of linear predictor analysis is to determine a set of predictor coefficients α_k in such a manner as to obtain a good estimate of the spectral properties of the speech signal through the use of equation (2.14). A basic approach is to find a set of predictor coefficients to minimise the mean square error over a short segment of speech. The resulting parameters are then assumed to be the parameters of the system function $H(z)$ in the model of Figure 2.5.

The short time prediction error average is defined as

$$E_n = \sum_m e_n^2(m) \quad (2.15)$$

$$= \sum_m (x_n(m) - \hat{x}_n(m))^2 \quad (2.16)$$

$$= \sum_m x_n(m) - \sum_{k=1}^p \alpha_k x_n(m-k)^2 \quad (2.17)$$

where $x_n(m)$ is a segment of speech selected in the vicinity of sample n , and m is temporarily left unspecified.

The values α_k can be found so as to minimise E_n of equation (2.17), by setting

$$\partial E_n / \partial \alpha_i = 0, \quad i = 1, 2, \dots, p$$

thereby obtaining the set of equations

$$\sum_m x_n(m-i) x_n(m) = \sum_{k=1}^p \alpha_k \sum_m x_n(m-i) x_n(m-k)$$

$$1 \leq i \leq p \quad (2.18)$$

From Equations (2.17) and (2.18) the minimum mean squared error can be shown to be

$$E_n = \sum_m x_n^2(m) - \sum_{k=1}^p \alpha_k \sum_m x_n(m) x_n(m-k) \quad (2.19)$$

It is now time to specify the duration m . If m is considered finite then we obtain a cross correlation matrix for the coefficients of equation (2.19) which is called a covariance matrix and has its methods of solution [3, 5, 10].

However, we arrive at a simpler structured Toeplitz matrix if it is assumed that m is infinite and obtain the autocorrelation matrix for the coefficients of equation (2.19).

2.2.5.1 Autocorrelation method [4]

Here it is assume that m for equation (2.19) is infinite. Defining the autocorrelation of $x_n(m)$ as

$$R_n(i) = \sum_{m=-\infty}^{\infty} x_n(m) x_n(m-i) \quad (2.20)$$

Equations (2.18) and (2.19) respectively reduce to

$$\sum_{k=1}^p \alpha_k R_n(i-k) = R_n(i), \quad 1 \leq i \leq p \quad (2.21)$$

$$\text{and } E_n = R_n(0) + \sum_{k=1}^p \alpha_k R_n(k) \quad (2.22)$$

Since the coefficients $R_n(i - k)$ of equation (2.21) form an autocorrelation matrix, we derive the "autocorrelation method".

In matrix form equation (2.21) can be expressed as

$$\begin{bmatrix} R_n(0) & R_n(1) & \dots & R_n(p-1) \\ R_n(1) & R_n(0) & \dots & R_n(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ R_n(p-1) & R_n(p-2) & \dots & R_n(0) \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ \vdots \\ p \end{bmatrix} = \begin{bmatrix} R_n(1) \\ R_n(2) \\ \vdots \\ R_n(p) \end{bmatrix}$$

This $p \times p$ matrix is a Toeplitz matrix, the solution of which may be efficiently done by the Levinson Recursion as modified by Durbin [3, 4].

As an additional consideration, a Toeplitz matrix is guaranteed to be non singular and hence the resulting all pole filter. However, in order to implement the autocorrelation method for short time speech segments we use an appropriate window function $w(n)$ so that another signal $x'_n(m)$, that is zero outside some interval $0 \leq m \leq N-1$, is obtained. Then the autocorrelation method can be applied to this $x'_n(m)$.

2.3 DURBIN'S RECURSIVE SOLUTION

In the preceding section a Toeplitz matrix was obtained for the autocorrelation coefficients of equation (2.21). The procedure can be stated as follows:

$$E_n^{(0)} = R_n(0) \quad (2.23)$$

$$k_i = [R_n(i) - \sum_{j=1}^{i-1} \alpha_j^{(i-1)} R_n(i-j) / E_n^{(i-1)}] \quad 1 \leq i \leq p \quad (2.24)$$

$$\alpha_i^{(i)} = k_i \quad (2.25)$$

$$\alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)} \quad 1 \leq j \leq i-1 \quad (2.26)$$

$$E_n^{(i)} = (1 - k_i^2) E_n^{(i-1)} \quad (2.27)$$

Equation (2.24) - (2.27) are solved recursively for $i = 1, 2, \dots, p$ and the final solution is given as

$$\alpha_j = \alpha_j^{(p)} \quad 1 \leq j \leq p \quad (2.28)$$

It is observed that in the process of solving for the predictor coefficients of a predictor of order p , the solutions for coefficients for predictors of lesser order have been obtained.

Since, at each iteration, we obtain $E_n^{(i)}$, it is easy to examine the error as the order increases. The set of intermediate parameters k_i obtained are called "reflection coefficients" and, in fact, corresponds to the reflection coefficients at the boundaries between successive sections of an acoustic tube with sections of fixed length and varying cross sectional area.

2.4 SYNTHESIS METHODS

Except in the case of homomorphic and short time spectral analysis, two basic synthesizers are used for

synthesis of speech from the analysis information:

- a) Terminal analog synthesizer
- b) Acoustic tube analog synthesizer.

2.4.1 Terminal Analog Synthesizer

Such a synthesizer [1] is directed at implementing a system whose transfer function approximates the vocal tract transfer function, but whose implementation bears no direct relation to the details of a vocal tract. The representation is only from a terminal view point. The basis of implementation is that its transfer function can be approximated by a cascade combination of resonant circuits, each one representing one of the modes of the vocal tract. Hence this type of a synthesizer is also called a "formant synthesizer" and its general structure is depicted in Figure 2.7.

To correspond to changes in resonance due to change in the shape of the vocal tract, the resonant circuits are provided with a set of time varying parameters to control the centre frequency and bandwidth of the resonators. A source shaping filter provides appropriate spectral coloration when the excitation used for voiced speech is an impulse train and white noise for unvoiced speech. In addition, a filter that accounts for the effect of the coupling of the acoustic tube into space is required.

2.4.2 Acoustic Tube Analog Synthesizer

Figure 2.8 shows the representation of the vocal tract as an acoustic tube [1, 5] consisting of a set of

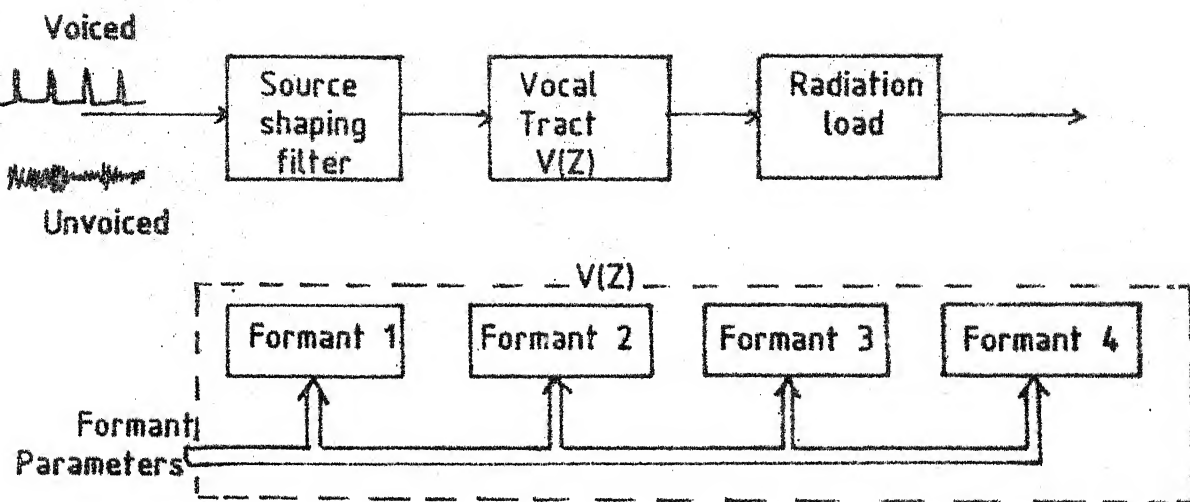


Fig. 2.7 Terminal Analog Synthesizer

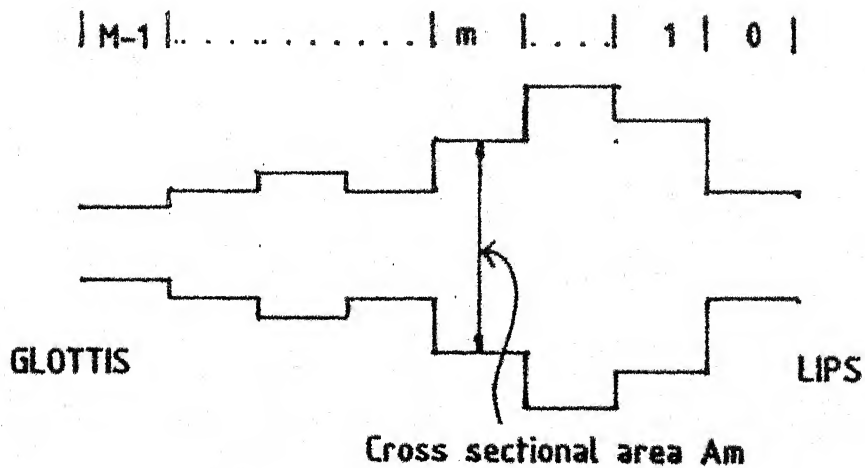


Fig. 2.8 Acoustic Tube model of the Vocal tract

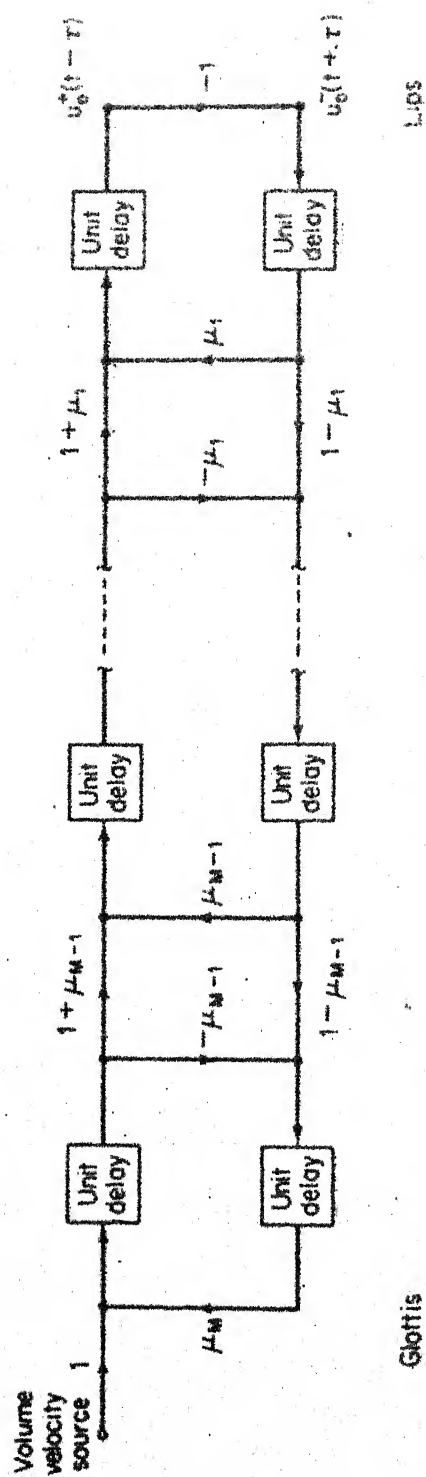


FIGURE 2.9 Linear signal-flow graph depicting the relationships between the forward and reverse traveling volume velocity waves throughout the acoustic tube model

interconnected sections of equal length and varying cross sectional area. The assumptions made are:

(a) The sound propagation through each section can be treated as a plane wave.

(b) The internal losses and the effect of the nasal tract and coupling between the vocal tract and the glottis can be ignored.

These assumptions permit an analysis of the acoustic tube model which leads to a filter structure whose variables are related to the physical variables of the tube. The end result is the linear flow graph structure of Figure 2.9. This flow graph depicts the relationships between the forward and reverse travelling volume velocity waves throughout the acoustic tube of Figure 2.8. The coefficients r_n , $n = 1, 2, \dots, M-1$ are the reflection coefficients that characterise the sound produced at the lip end of the tube.

2.5 VOCODERS

Having seen the various analysis and synthesis methods available, vocoders based on these methods are reviewed at block diagram level.

- a) Channel vocoder [17]
- b) Homomorphic vocoder [3, 10, 11]
- c) Vocoder based on Formant analysis [1, 14]
- d) LPC vocoder [15, 18].

2.5.1 Channel Vocoder

The channel vocoder implements a short time spectral analysis on the speech segments as shown in Figure 2.10(a) and a synthesiser that is peculiar to this analysis method is shown in Figure 2.10(b). The filter bank method of analysis is used. It is pertinent to point out that these filter bands are desired to be wideband so as to yield a smoother spectrum (Figure 2.10(c)), since the wider filters average over several harmonics of the fundamental frequency.

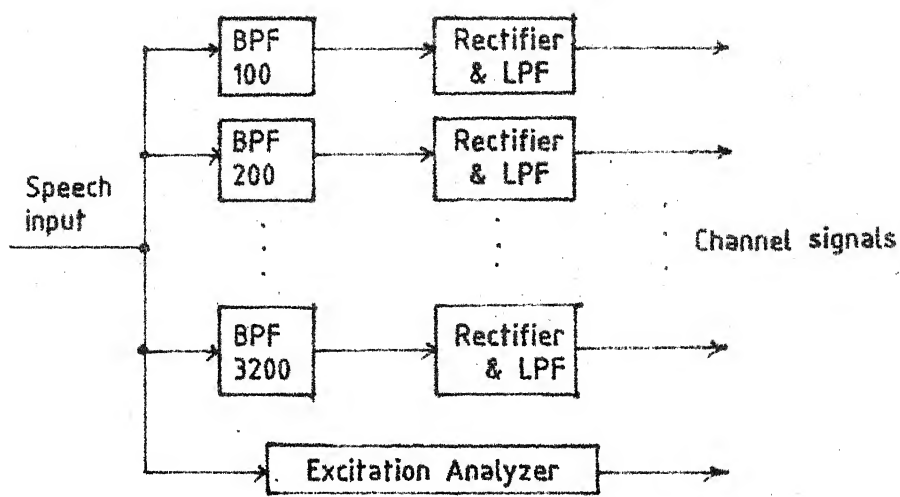
The number of filters is generally between 16 and 20, and the low pass smoothing following rectification is chosen to suppress the pitch ripple of 50 Hz and higher while passing the vocal tract spectral changes [11]. The pitch may be detected by any one of the several methods in Chapter 3.

2.5.2 Homomorphic Vocoder

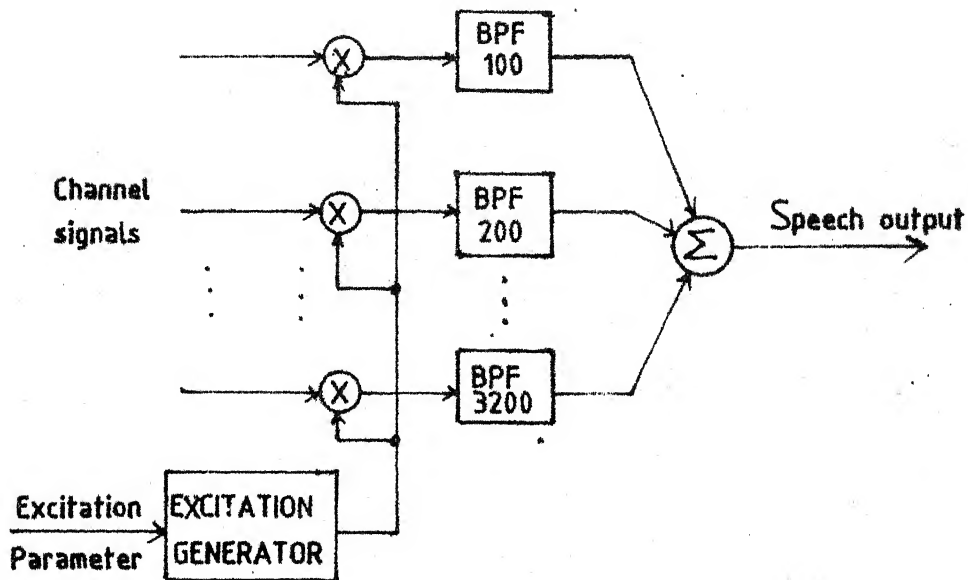
The analyser and synthesizer for a homomorphic vocoder are shown in Figure 2.11. Here again the synthesizer is peculiar to the analysis method used. The vocal tract transfer function $V(nT)$ is obtained from the windowed cepstrum, as suggested earlier in Section 2.2.3. This is then convolved with the excitation parameters to obtain synthesised speech.

2.5.3 Vocoders Based on Formant Analysis

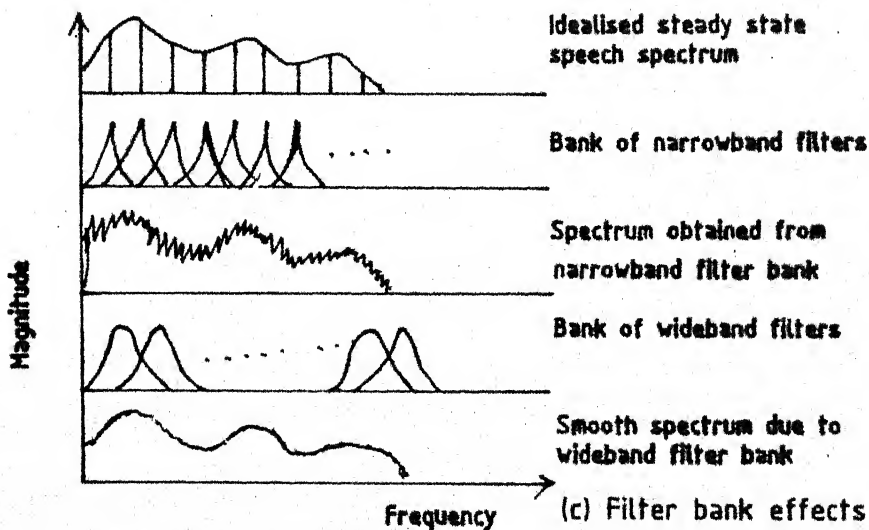
In Section 2.2.4 it has been shown how formant frequencies can be extracted from a speech segment. Studies



(a) Analyzer

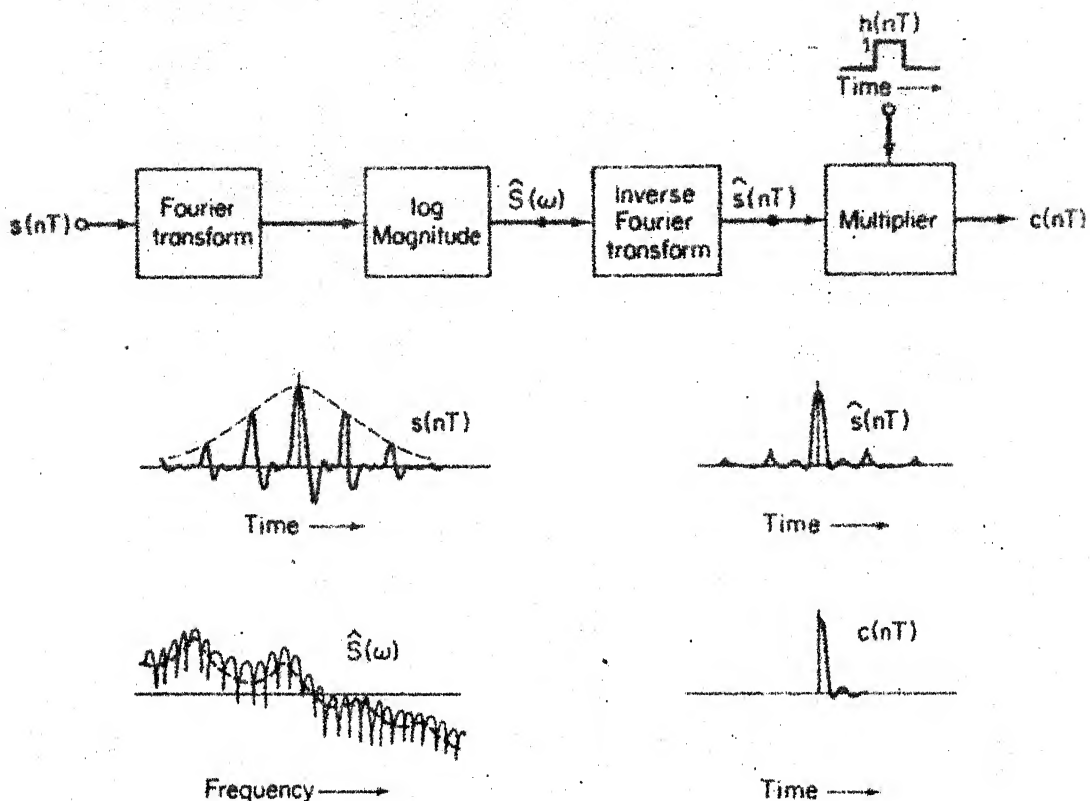


(b) Synthesizer

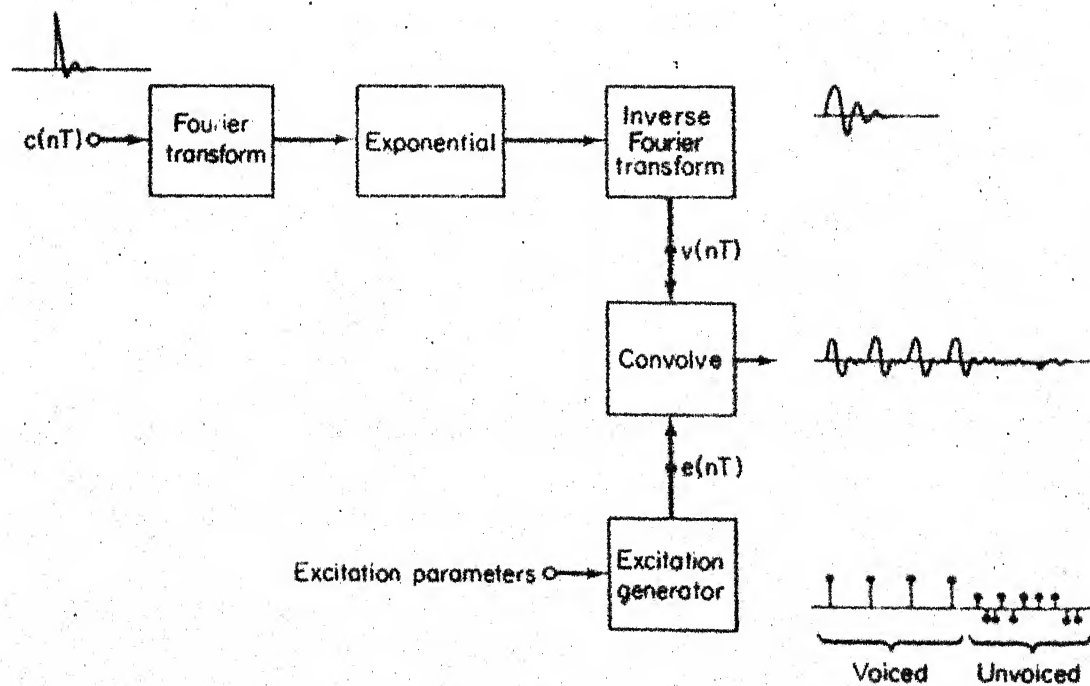


(c) Filter bank effects

Fig 2.10 CHANNEL VOCODER



(a)



(b)

FIGURE 2.11 Block diagram of homomorphic analysis-synthesis system; (a) analyzer configuration, (b) synthesizer configuration.

show that the first three or four formants are enough to reproduce the speech in a terminal analog synthesiser.

2.5.4 LPC Vocoder

Figure 2.12 shows the major blocks needed to implement a LPC vocoder [18]. The predictor coefficients are not really extracted for use in synthesis here. Rather, the reflection coefficients, which are generated in the process of implementing recursive algorithms to solve the linear equations for predictor coefficients, are used for synthesis using an acoustic tube synthesiser. This follows from the fact that these reflection coefficients are found to correspond to the reflection coefficients at the boundaries of the fixed length sections of an acoustic tube.

Such an implementation using the acoustic tube overcomes the problems [3] of using a direct form structure generated directly from the predictor coefficients, or a cascaded form obtained from factoring the all pole transfer function

$$H(z) = \frac{G}{1 - \sum_{k=1}^p \alpha_k z^{-k}}$$

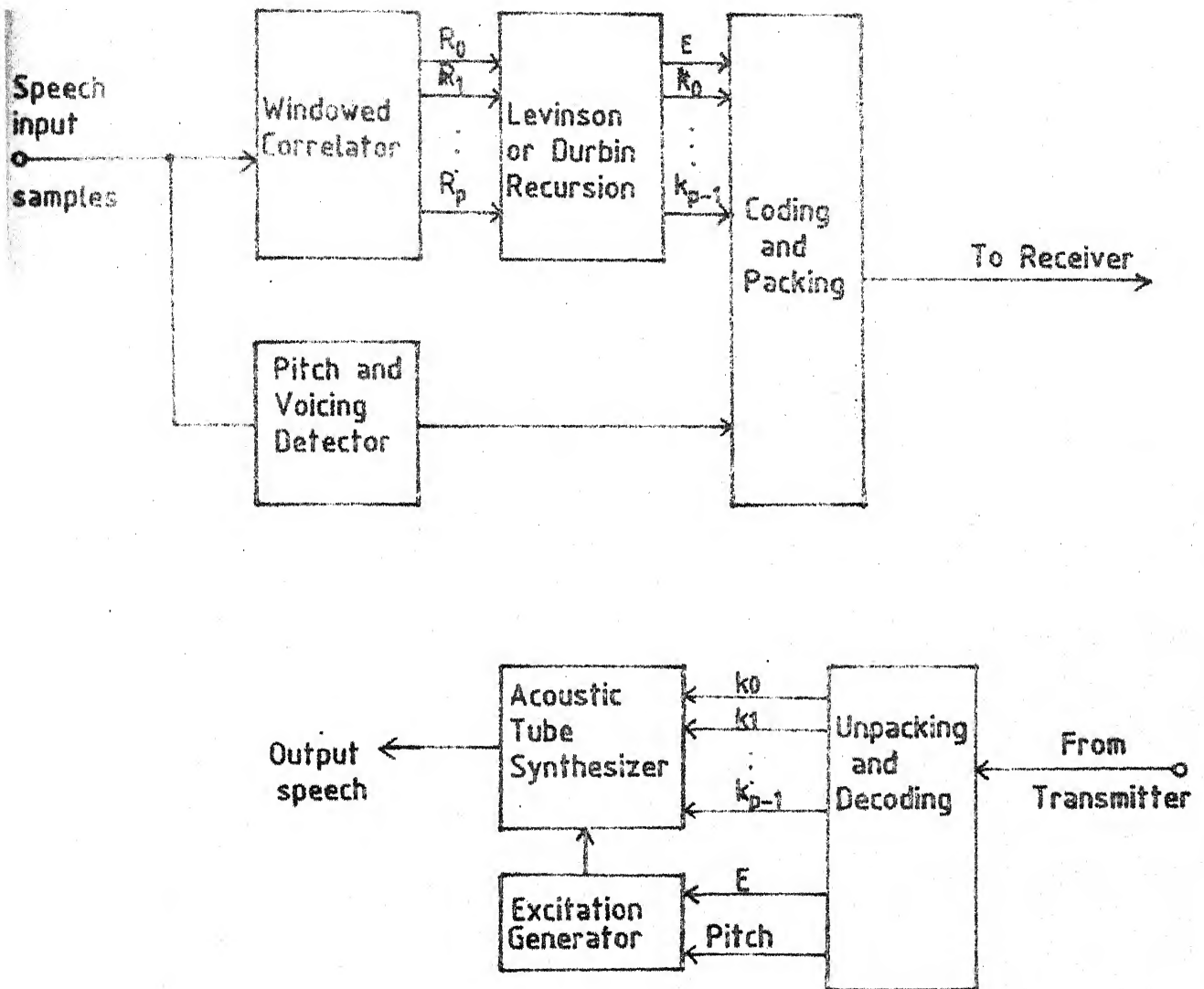


Fig 2.12 A LPC Vocoder

2.6 SUMMARY

This chapter reviewed the various analysis-synthesis systems available. Further work revolves around the implementation of the LPC vocoder since it involves simpler hardware and computation schemes. The excitation function which is sought to be separated in all the cases, is derived from a number of pitch period estimators mentioned in Chapter 3.

CHAPTER 3

EXCITATION ANALYZERS

3.1 PITCH PERIOD ESTIMATION

Any analysis of the speech signal must provide, as the result, the vocal tract function and the excitation function separately, in one form or another. To this end a number of efficient pitch period estimation algorithms have been developed. All the algorithms provide the two distinct pieces of information that constitute the excitation function.

- (a) Whether a particular speech segment is voiced or unvoiced and
- (b) if voiced, the pitch period of periodic excitation for that segment.

The pitch period estimation algorithms available are:

- 1) Modified autocorrelation method using clipping (AUTOC)
(Figure 3.1)
- 2) Cepstrum method (Figure 3.2)
- 3) Simplified inverse filtering technique (SIFT)
(Figure 3.3)
- 4) Data reduction method (DARD) (Figure 3.4)
- 5) Spectral equalisation LPC method using Newton's transformation (LPC) (Figure 3.5)
- 6) Average magnitude difference function (AMDF)
(Figure 3.6)
- 7) Parallel processing method (PPROC) (Figure 3.7).

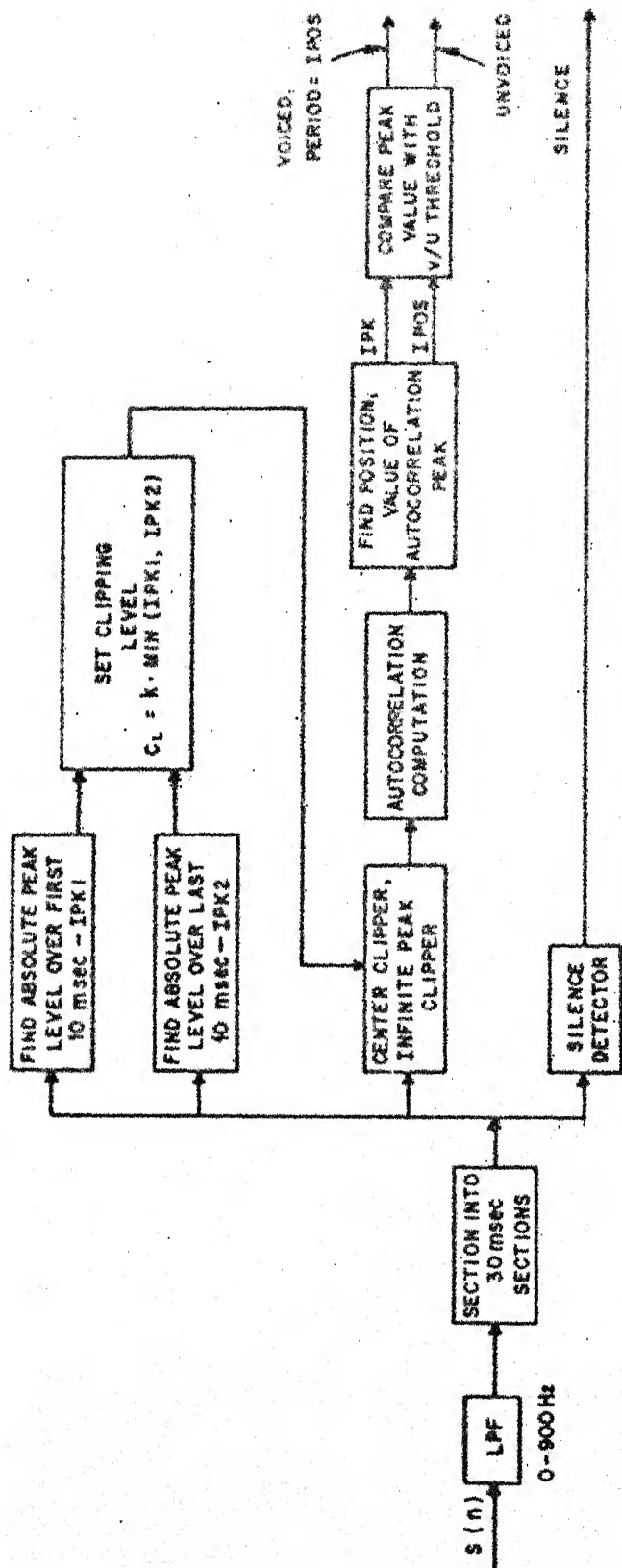


Fig 3.4 Block diagram of the AUTO C pitch detector.

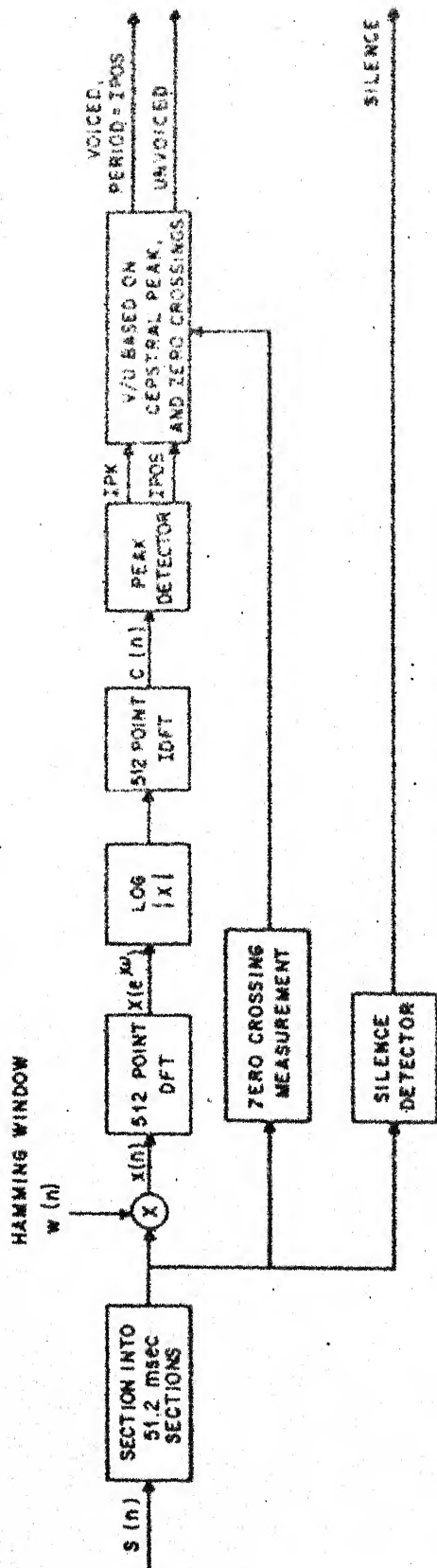


Fig 3.2 Block diagram of the CEP pitch detector.

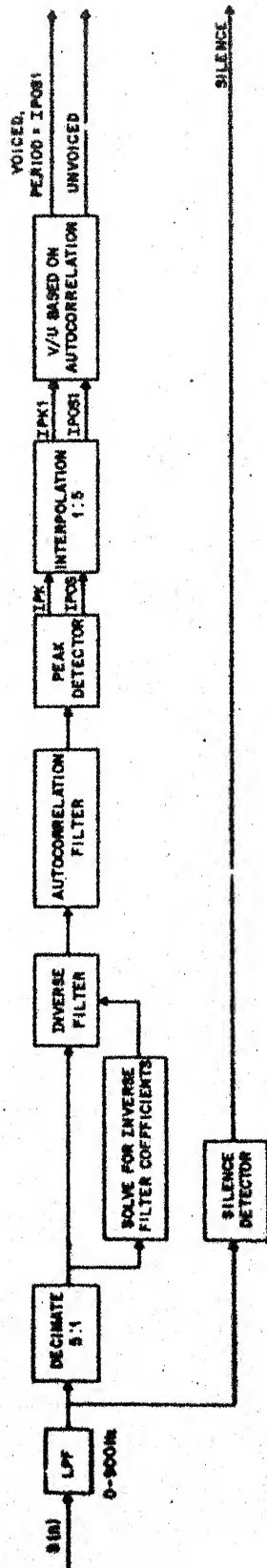


Fig. 3.3 Block diagram of the SIFT pitch detector.

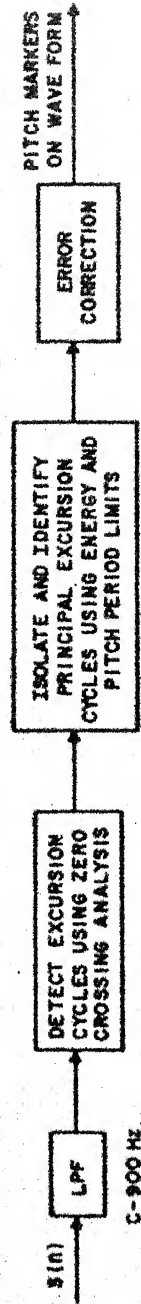


Fig. 3.4 Block diagram of the DARD pitch detector.

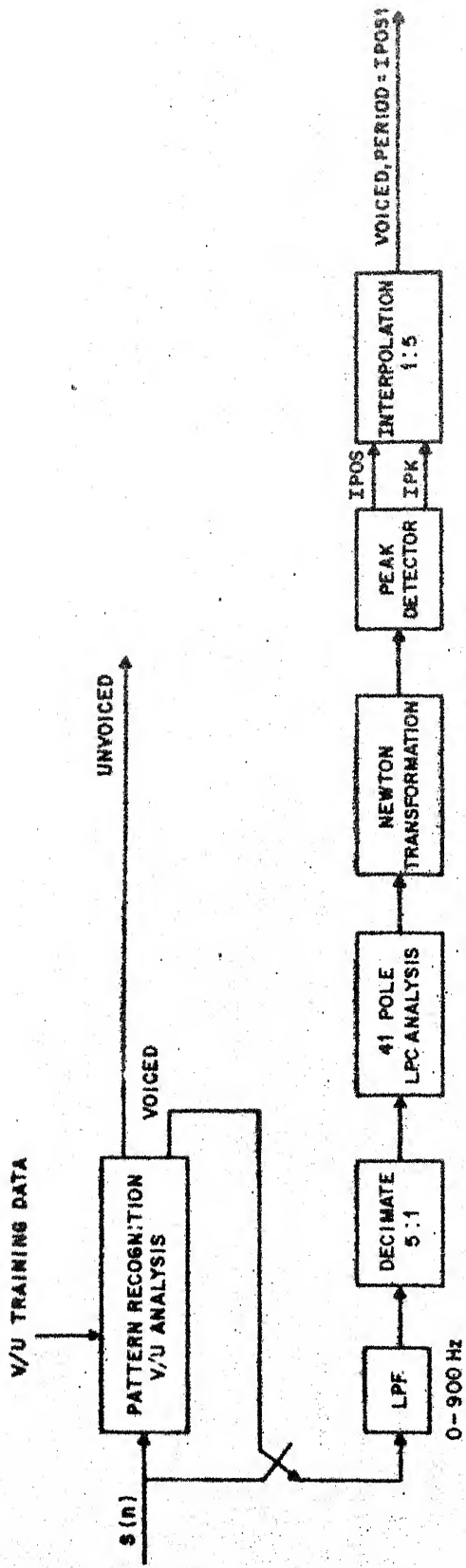


Fig 3.5 Block diagram of the LPC pitch detector.

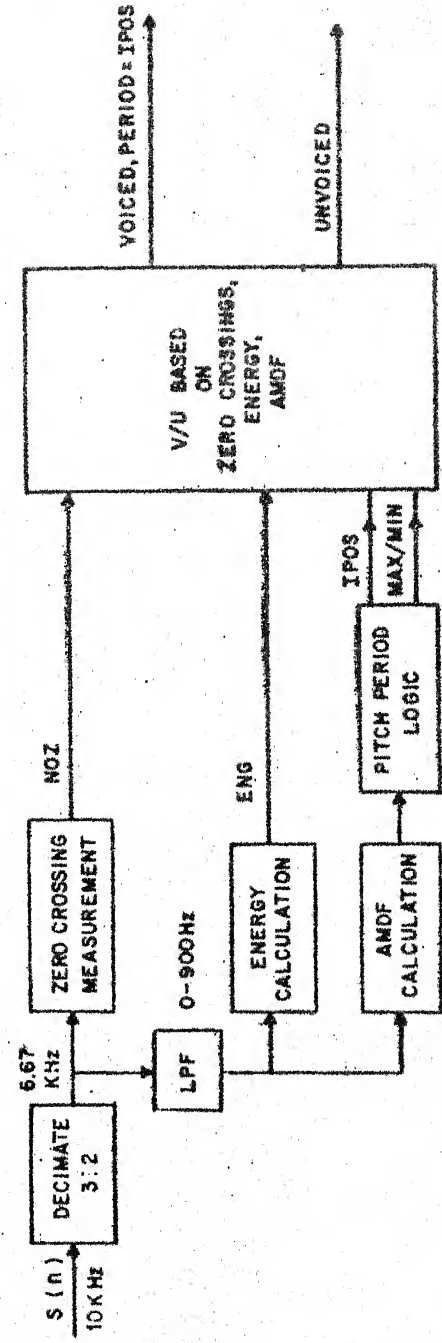


Fig 3.6 Block diagram of the AMDF pitch detector.

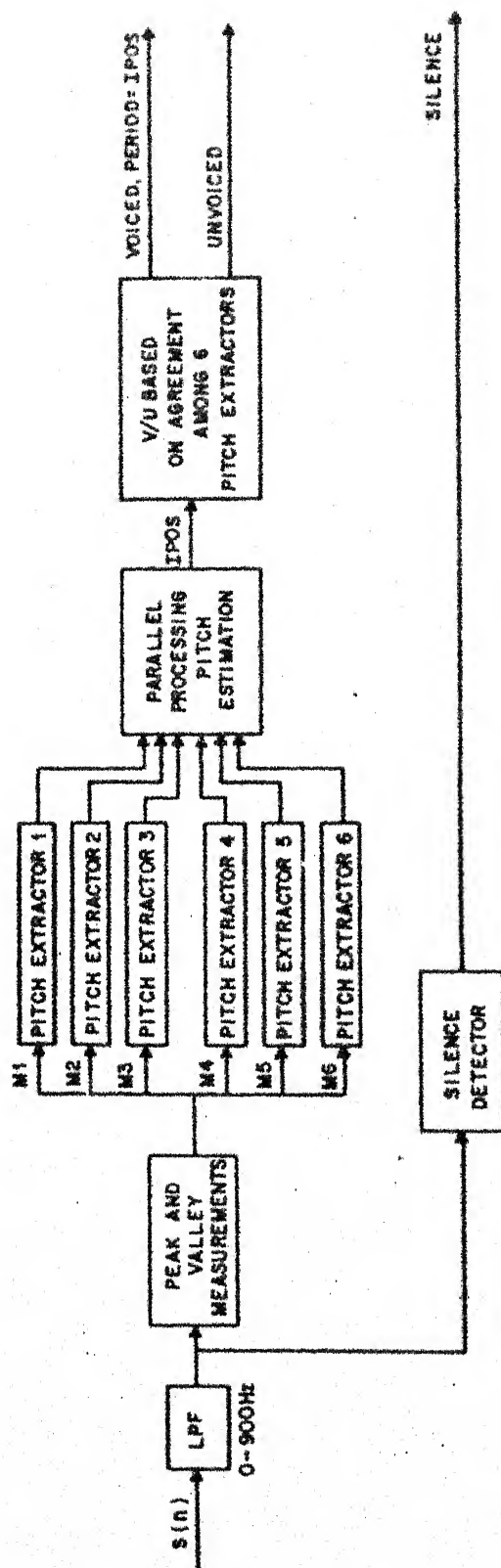


Fig 3.7 Block diagram of the PPROC pitch detector.

3.2 SELECTION FOR IMPLEMENTATION

The selection of an algorithm for implementation from amongst the methods mentioned was made from a study [13] based on the following criteria:

- 1) Accuracy in estimating pitch period
- 2) Accuracy in making a voiced-unvoiced decision
- 3) Robustness of the measurements, i.e. they must be modified for different transmission conditions, speakers etc.
- 4) Speed of operation
- 5) Complexity of the algorithm
- 6) Suitability for hardware implementation
- 7) Cost of hardware implementation.

Based on this study, it was decided to implement the Parallel processing algorithm (PPROC) of Gold and Rabiner for the following reasons:

- 1) The parallel processing involved, helps produce fairly accurate pitch period estimates over a range of speakers.
- 2) The algorithm is relatively straightforward and employs simple signal processing.
- 3) A real time implementation was found feasible using off-the-shelf components.

3.3 THE PARALLEL PROCESSING ALGORITHM

Gold and Rabiner [6] proposed a means of obtaining pitch period estimates by parallel processing of the peak and valley measurements made on the incoming speech signal

(Figure 3.7). These values, named M_1 to M_6 are fed to six independent pitch extractors or pitch period estimators. The estimates from these extractors, obtained parallelly, are put through a sophisticated comparison algorithm with a bias. The final decision of voiced and unvoiced segments is made based on the agreement among the six pitch extractors in the above comparison. For voiced segments, the most popular candidate in the comparison is put out as the pitch period.

However, to reduce the complexity of the peak and valley measurements, a modified version of this algorithm, has been implemented.

3.4 THE IMPLEMENTED MODIFICATION

The modification, called the **second modification** to the original parallel processing algorithm is discussed below. The block diagram is in Figure 3.8. The basic differences from the original algorithm are discussed after pointing out that this modification is made on an assumption that fundamental frequencies are expected to be below 300 Hz. Based on this assumption the changes are:

- (1) The peak and valley measurements have been replaced by selective filtering through two Lerner filters with bandwidths 80-240 Hz and 200-600 Hz as shown. Further, the filtered signals are fed into positive and negative peak detectors. The outputs of these detectors are positive pulses corresponding to the signal peaks, and whose magnitude is also equal to the peak magnitude of the signal at that point.

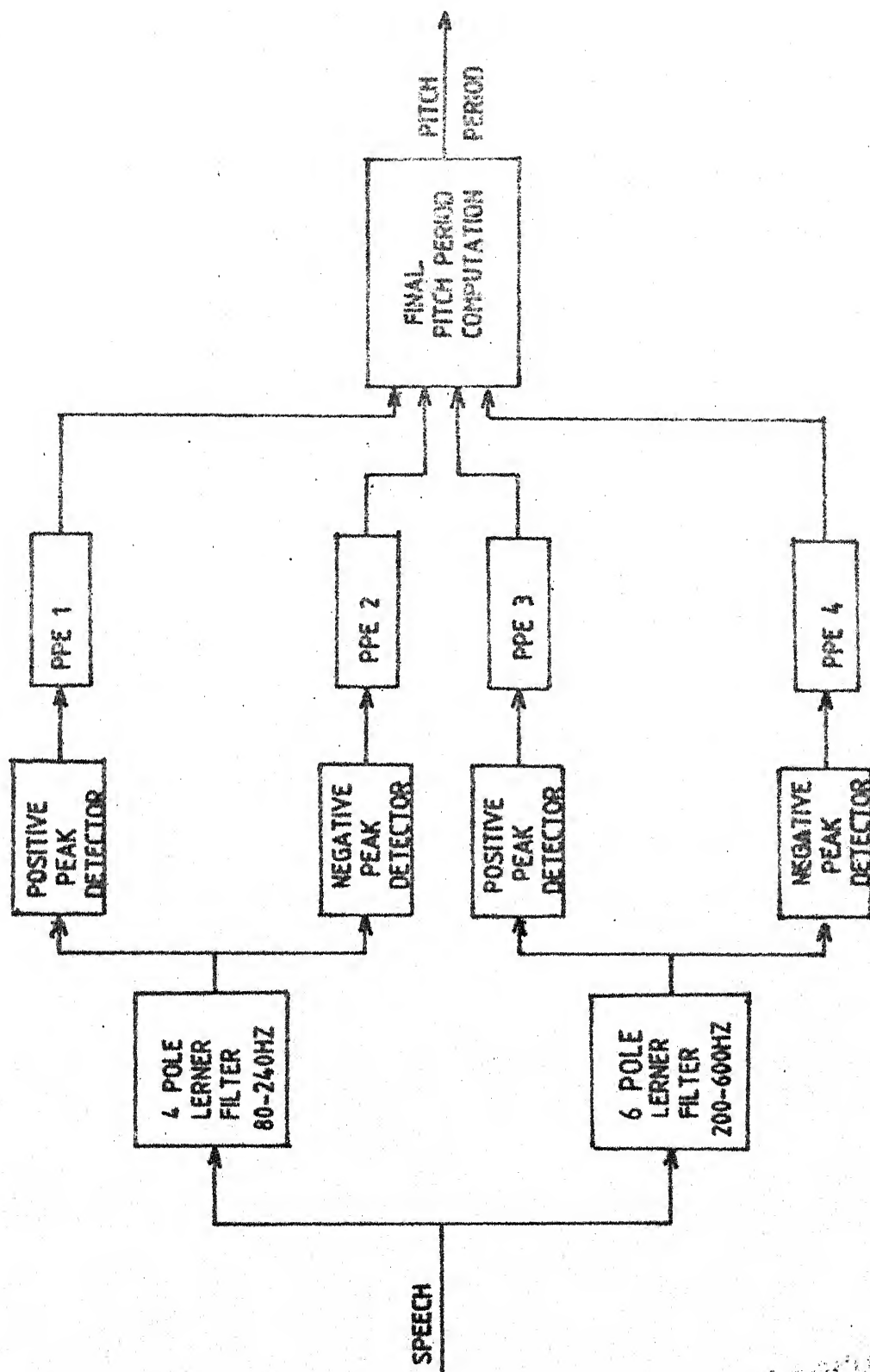


Fig 3.8 Block Diagram of the Pitch Period Estimation Algorithm

87539

(2) The number of pitch extractors or pitch period estimators (PPE) has been reduced to four, one for each negative and positive peak detector. The detector consists of an exponential run down circuit which is set to the value of the peak. Once set a blanking period follows during which peaks if any are ignored. At the end of the blanking duration, the circuit is allowed to run down. If during this time any pulse comes which is greater in magnitude than the current run down value, then the run down circuit is reset to that value and the time interval elapsed between the two resets is put out as the pitch period (Figure 3.9). The original algorithm employed a variable blanking time and run down time constant based on the last pitch period average. In the implemented modification both these are fixed:

- a) The blanking intervals is fixed to 2 ms.
- b) The run down time constant is such as to run down to half the initial value in 5 ms.

In addition the run down circuit is reset after 16 msec if no reset occurs, indicating an unvoiced segment.

(3) The bias used in the original algorithm is removed and the final computation is performed by a coincidence check on the table formed in Figure 3.10. The coincidence measurements are conducted for the four most recent estimates P_i (P_{11} , P_{21} , P_{31} , P_{41}). Each of these "candidates" are compared with the other eleven (P_c) and a score is maintained for each P_i based on the inequality

$$|P_c - P_i| < \frac{1}{8} P_i \quad (3.1)$$

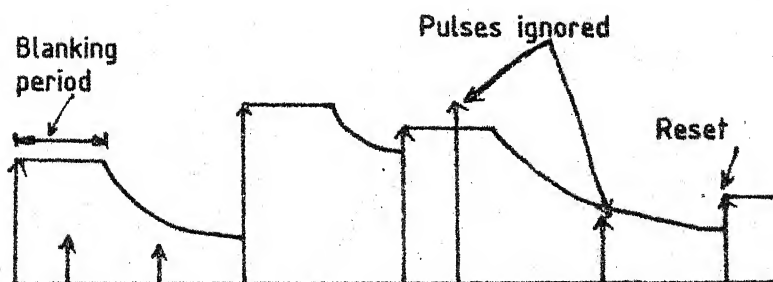
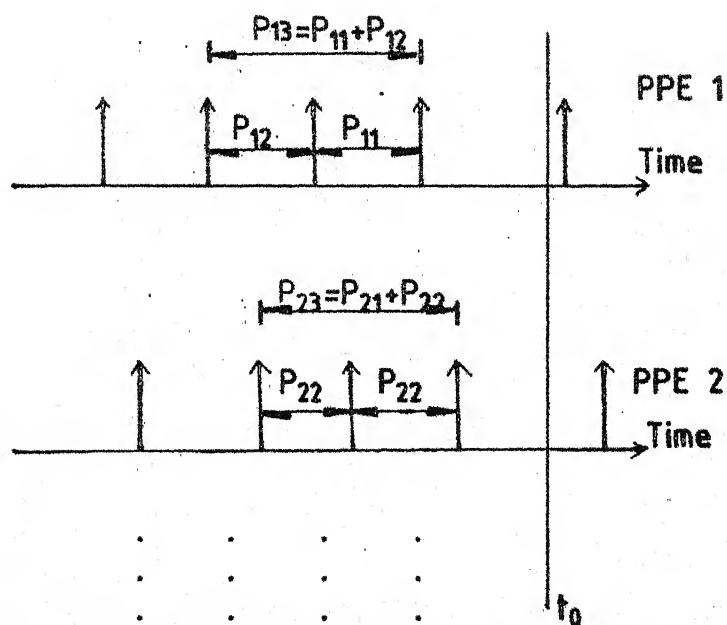


Fig 3.9 Run Down circuit



PPE NUMBER

P_{11}	P_{21}	P_{31}	P_{41}
P_{12}	P_{22}	P_{32}	P_{42}
P_{13}	P_{23}	P_{33}	P_{43}

Estimated
Pitch
Periods

Fig 3.10 Final Pitch Period Computation

The voiced-unvoiced decisions are then made on the following basis:

- a) The candidate scoring maximum coincidences is put out as the pitch period.
- b) If any of the two candidates are 16 msec or more than then an unvoiced or "hiss" decision is made.
- c) Also if no candidate gets at least 4 "votes", a hiss decision is made.

Gold and Rabiner [6] have indicated that for the assumption that only fundamentals below 300 Hz are expected, the performance of the modified algorithm is reasonably good.

CHAPTER 4

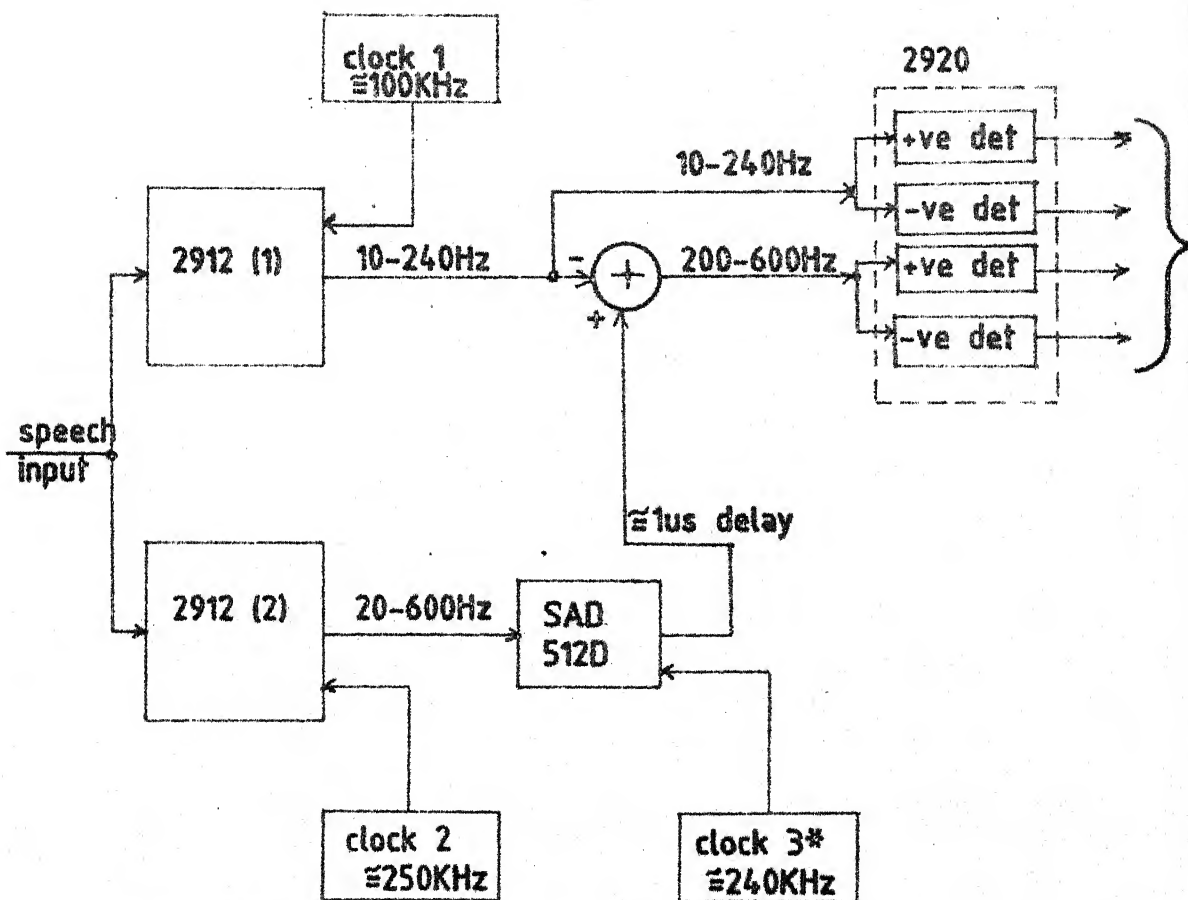
IMPLEMENTATION OF THE PARALLEL PROCESSING PITCH PERIOD ESTIMATOR

4.1 CONSIDERATIONS FOR THE DESIGN

The basic consideration for the design of the circuit and software for the implementation of PPROC algorithm has been: the use of off-the-shelf components. It was decided to use the 2912 line filter and the 2920 signal processing chips to implement the major signal processing functions. An initial attempt was made to realise the two band pass filters using the 2912 line filters with 4 software peak detectors residing in the 2920. However this was turned down subsequently in favour of a final implementation in which the filters, peak detectors and the run down circuits were implemented on two 2920 ICS thereby drastically reducing the chip count for the overall system.

4.2 THE INITIAL ATTEMPT

Based on the consideration that the cut off rates visualised by the 4 pole and 6 pole Lerner filters [23] could be equalled by the 2912 line filter, the scheme shown in Figure 4.1 was tried. A linear relationship was found to exist between the clock frequency to the 2912 filter and its cut off frequency. This property was used to derive the two band pass filters with pass bands as shown. The SAD 512D analog delay IC was used to compensate for the



*clock adjusted to obtain correct delay
 &all outputs are positive pulses corresponding to peaks

Fig 4.1 Schematic of first attempt

additional delay in the signal through 2912(1). The 4 peak detectors were implemented in the 2920 and overall, the scheme worked upto this point.

However, further implementation of the run down circuits and blanking intervals for the pitch period estimators was found to need a large number of components per channel. Hence this attempt was dropped and a more elegant final circuit was derived.

4.3 THE FINAL IMPLEMENTATION

The initial attempt ran into problems when processing after peak detection was considered. However, rethinking suggested a better implementation resulting in easier control of the run down circuits and a reduced chip count. The hardware and software aspects are separately discussed in the following sections. The final circuit was a single card containing the ICS as shown in Appendix A-1. This card provides the 4 parallel estimates of the pitch period and is plugged into the available workstation to perform the final computation.

4.3.1 The Hardware

A detailed circuit diagram is available in Appendix A-2 read with Appendix A-3.

Assuming speech input from an ordinary telephone handset, a two-stage, front end amplifier using two 741s (U8, U9) has been implemented for the necessary gain. The speech signal is then passed through the 2912 line filter

with the upper cut off frequency adjusted to 600 Hz using an NE555 clock (U1). This signal is passed to the SIGINO pins of the two 2920 ICS (U11, U12).

4.3.1.1 Use of 2920 signal processor

The pin layout is presented in Figure 4.2 for a 2920 IC 21. Notes on the 2920 signal processor and the use of the 2920 signal processing software package are appended at Appendix D. The two 2920 ICS (U11, U12) have been used in an identical manner. Each is provided with a 6.144 MHz crystal which implies a 8 KHz sampling rate when the full program length of 192 instructions is used. The reference voltage for A/D and D/A conversion is provided by the set up of one LM 308 and two LM 103 voltage reference ICS (U2, U3, U4). At the processor output pins the mode of output is controlled by the mode control pins 25 (M_1) and 24 (M_2). By fixing M_1 at -5 volts and M_2 at +5 volts, the sigont pins 0-3 are made to provide TTL outputs. When external pull-up resistors are connected to these pins each has an output compatible to one TTL gate.

Inside U11 is the software implementation of a 4 pole Lerner filter, a positive and negative peak detector and two exponential run down circuits, one for each peak detector. The input signal is applied at pin 10 (signin 0) and is sampled by the internal A/D convertor (Figure 4.2) under program control. The input sample is propagated through the software digital filter within and peaks are detected in the software peak detectors. Signin 1 gets the

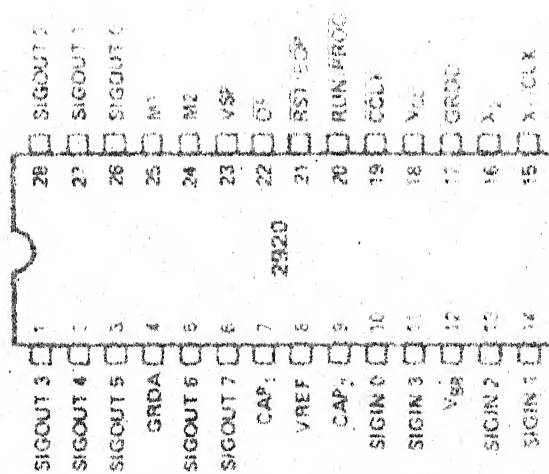


Fig 4.2 Pin layout of 2920

2 ms blanking pulse for the run down circuit corresponding to positive peaks (PPE1 of Appendix A-2). A TTL level pulse is output at pin Sigout 0 whenever the run down circuit is reset. Similarly sign 2 gets the 2 ms blanking pulse for the negative peak run down circuit (PPE2 of Appendix A-2) and a TTL level pulse appears at pin sigout 2 whenever this circuit is reset.

U12 is similarly configured with the difference that a 6 Pole Lerner filter is implemented within.

Pin Sign 3 is provided with -2 volts so that when sampled it serves to discharge the external sample and hold capacitor rapidly enough to prevent error in reading the input values on pins Sign 1 and Sign 2. Sampling pins sign 1 and 2 indicates whether the blanking pulses are present or not.

The outputs of the 2920 ICS are buffered using 4 of the 6 buffers in a 7407 (U33). These 4 buffered pulses go one to each of the PPEs 1 to 4.

4.3.1.2 The pitch period estimators

Four identical blocks, named PPE 1 to 4, perform the functions of

- a) Providing a 2 ms blanking interval during which no run down occurs in the 2920.
- b) Counting the time interval between two successive pulses from the 2920s.
- c) A pulse is output from the PPE to indicate that 16 ms

or more have elapsed between two successive reset pulses of the run down circuit.

The detailed circuit of a PPE is in Appendix A-3. PPE 1 will be discussed in detail to represent all the PPEs.

The buffered pulse from the 2920 sigout pin is used to perform the following operations:

- a) Trigger on the rising edge, one of the two monostables of 74123 (U15), to generate a 2 ms blanking pulse.
- b) Trigger on the falling edge, the second monostable whose outputs, Q and \bar{Q} , are applied to the reset and trigger inputs of the XR 2240 counter (U25) to start a fresh count.
- c) Strobe into the 8282 latch (U29), on the falling edge, the latest count reached by the XR 2240 counter, i.e. the latest pitch period estimate.

The XR 2240 counter is reset under the following conditions:

- No. 1 - When a count representing 16 msec is reached (i.e. when the 3 MSB are all 1), or
- No. 2 - A Q output from the monostable appears due to a 2920 output.

The reset pulse is obtained by a logical OR of outputs of the monostable and the 4 input NAND gate, 1/74 LS20 (U16). The output of the 4 input NAND gate is also used to trigger on the falling edge, one of the monostables in U5. The outputs Q , \bar{Q} of this monostable are input to the latch 8282 (U7) which is made transparent. Thus U now holds

information that a 16 ms count has been exceeded in the concerned PPE.

The 8282 (U29) data is placed on the data bus when a $\overline{\text{CS}}$ is given to the $\overline{\text{OE}}$ pin of the 8282. This $\overline{\text{CS}}$ is obtained under software control from the dual 1 out of 4 decoder 74 LS 155 (U23) which uses $\overline{\text{IO}}/\overline{\text{M}}$, A15, A14, A13 to decode the chip to be selected.

4.3.1.3 The interrupt set-up

In the previous sections it has been shown how the four 8282 latches (U29, U30, U31, U32) always hold the most recent pitch estimate. The fifth latch U7 holds the information as to which of the PPEs has exceeded 16 ms in the recent count. All this information is to be read into the microprocessor for final computation.

The microprocessor is interrupted by an RST 7.5 interrupt generated by the timer U13, every 5 ms, suitably inverted by 1/7400 (U14) to meet the workstation requirements.

The microprocessor puts out the pitch period information on the screen of the workstation for every other RST 7.5, i.e. effectively once for every 10 ms. This timing can be varied by varying the RC of timer U13 suitably.

4.3.2 The Software

The software used is implemented in two basic parts:

- 1) 2920 signal processor software including a Lerner filter, positive and negative peak detector and run down circuits for both positive and negative peaks (Appendices B-1 & B-2)

- 2) 8085 program for final pitch period computation
(Appendix C).

4.3.2.1 2920 Signal Processor Software

Printouts appended at Appendices B-1 & B-2 contain the programs burnt into the PROM areas of U11 and U12 respectively.

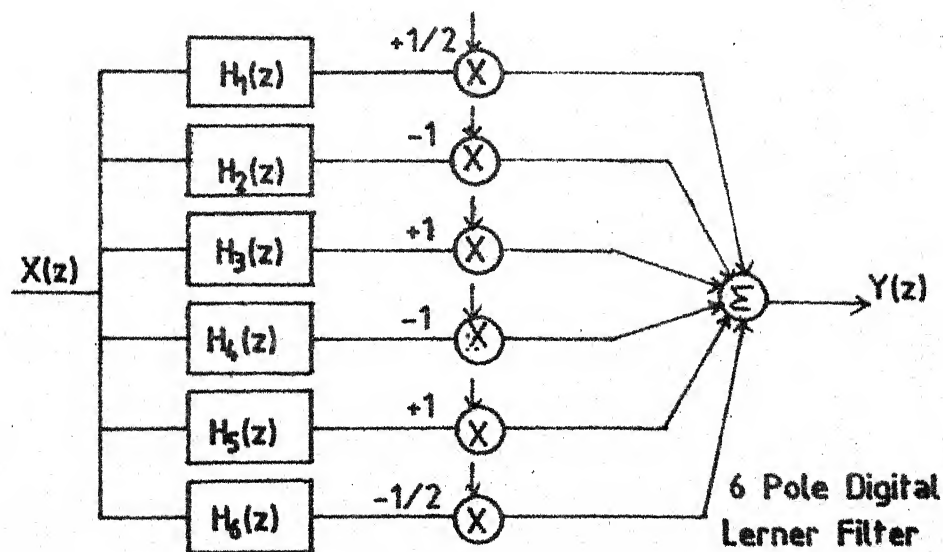
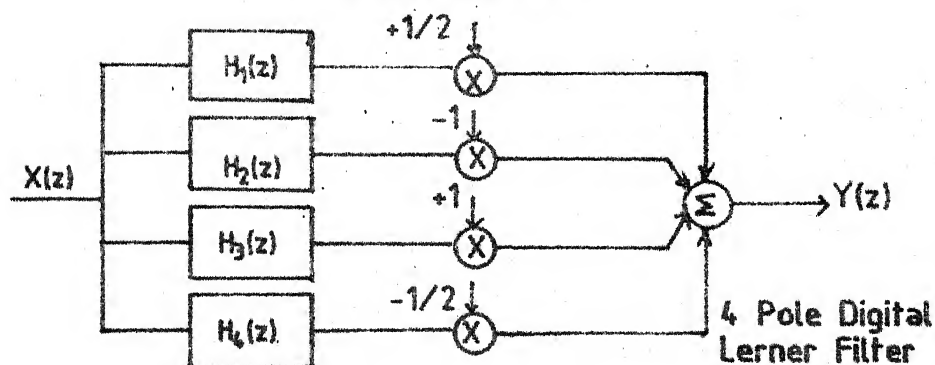
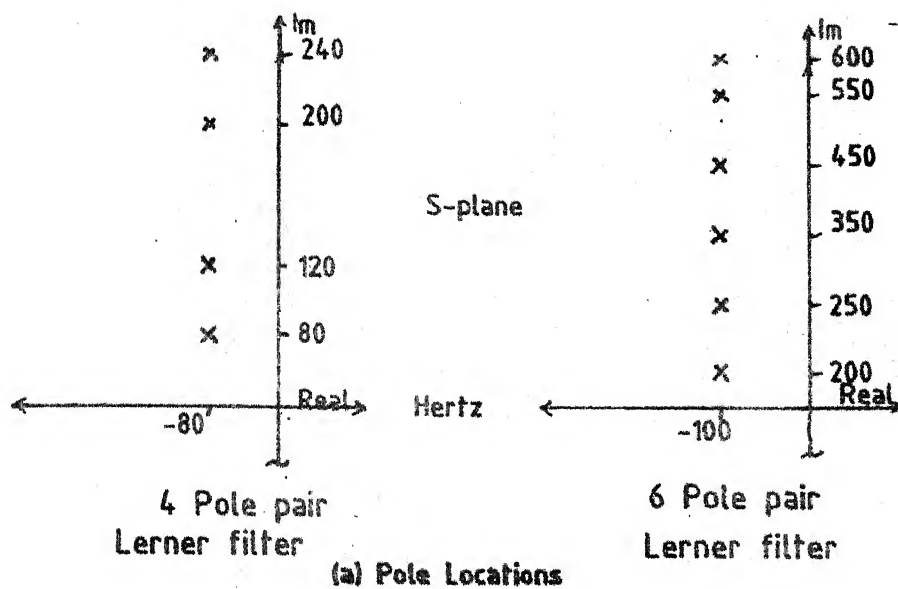
For U11, the 4 pole Lerner filter has complex pole pairs at 80, 120, 200, 240 Hz along the $j\omega$ axis, offset 80 Hz to the left of the imaginary axis.

For U12, the 6 pole Lerner filter has complex pole pairs at 200, 250, 350, 450, 550, 600 Hz along the $j\omega$ axis, offset 100 Hz to the left of the imaginary axis.

The pole locations and the Lerner filter implementations are in Figure 4.3(a) and (b) respectively. High cut off rates are realised for such filters which are constituted by summed outputs of parallel resonators [8, 23].

Extensive use of the SPAS 20 compiler [22] software available on the MDS-Inteltec 2 system has been used to develop the filters, taking care of the intermediate overflows. The output of each resonator is multiplied by an appropriate residue before summing as shown in Figure 4.3(b). The adjustment of the gains was aided by simulating the filter using the correct sampling rate with the help of the 2920 simulator [22] software also available on the MDS.

The rest of the program is easily understood by the flow chart in Figure 4.4. BIT A and BIT B are used to indicate whether or not a blanking pulse is present. If



(b) Digital implementation

Fig 4.3 Implementation of Digital Lerner Filters

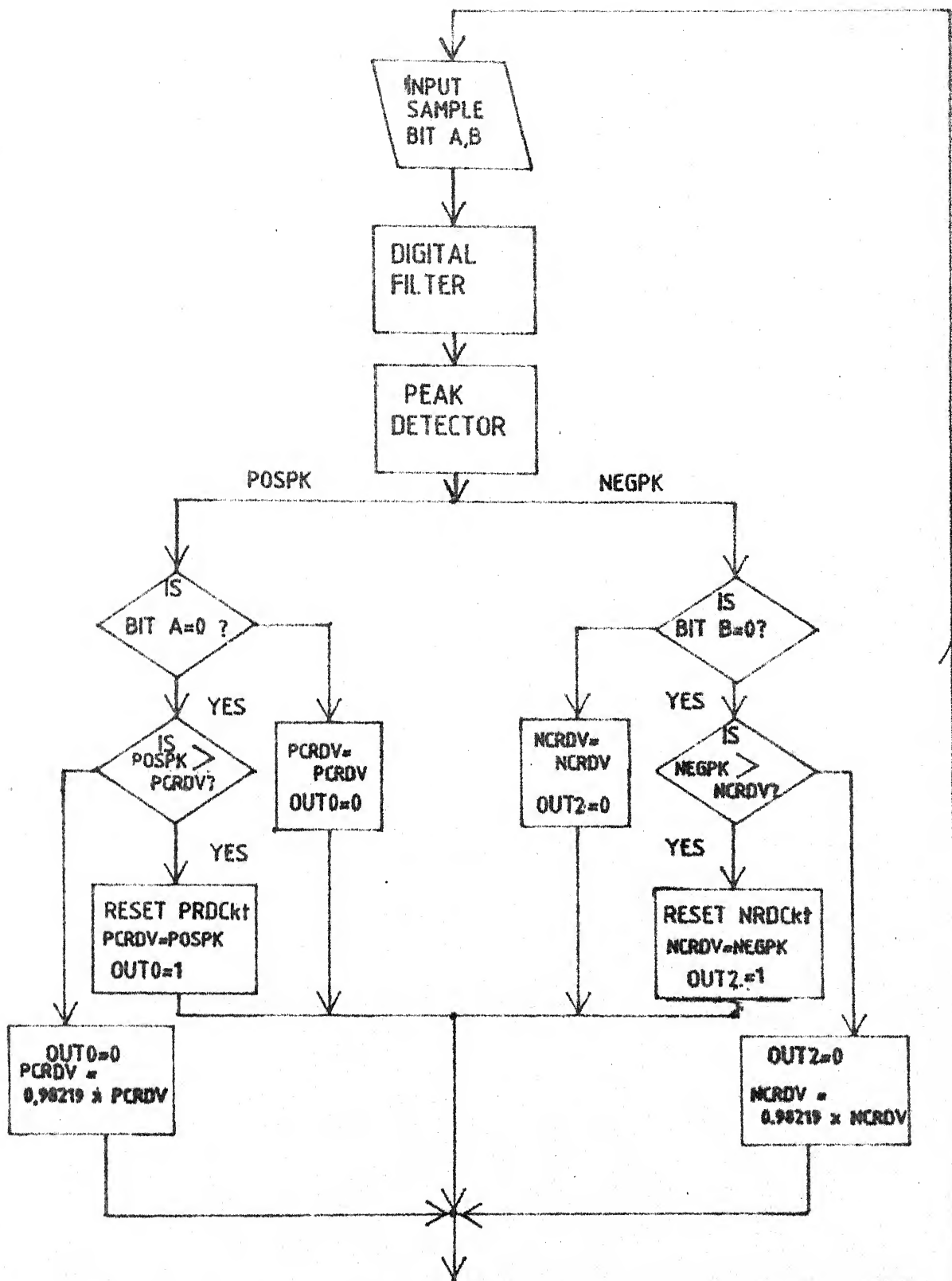


Fig 4.4 Flow Chart for 2920 Software

either is 1, then the respective run down circuit is updated with the same value as it has been set with. The run down is allowed to continue any time they drop to 0.

A single peak detector algorithm has been implemented which remembers the sign of the input signal and at the end gives out a positive or negative peak. This algorithm essentially compares 3 successive input samples and will put out as peak the centre value, when it is greater than the ones before and after it. This algorithm is shown in Figure 4.5.

For the run down circuit, the equation implemented is

$$y_n = 0.98219 y_{n-1}$$

and simulates an exponential run down with the desired time constant, i.e. run down to half initial value in 5 ms.

The analog output instructions OUT 0 and OUT 2 are used to output TTL level pulses at pins Sigout 0 and Sigout 2 corresponding to resets of the positive and negative run down circuits respectively.

4.3.2.2 Final computation software

Software for the final computation of the pitch period based on coincidence measurements for the latest 4 estimates P_{11} , P_{21} , P_{31} , P_{41} is implemented as a program for the 8085 and fed into the workstation.

A printout of the program is appended at Appendix C.

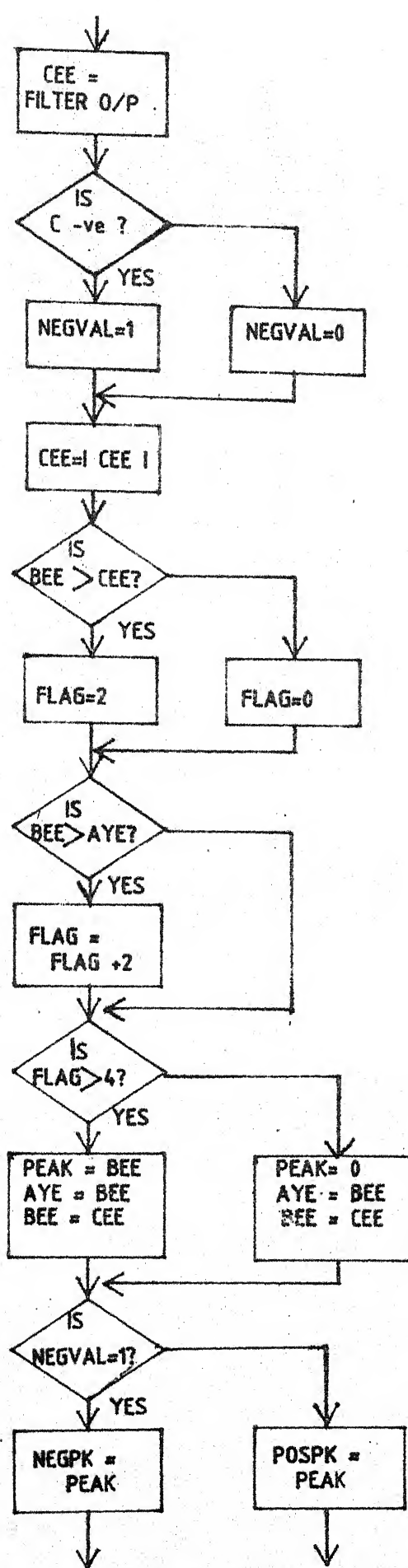


Fig 4.5 Flow Chart for PEAK DETECTOR

The program has two main parts:

- a) The background program, which currently wastes time but could be used for control and communications in a full vocoder implementation.
- b) The interrupt service routine which is entered each time RST 7.5 interrupt is caused by the timer on the card.

The ISS has two parts:

The first part is entered for the first RST 7.5 received. The reset status word of U7 and the values from the 4 estimators are read into the second row of the table of Figure 3.10, i.e. P_{12} , P_{22} , P_{32} , P_{42} . In case any estimator has indicated a reset of the counter caused by exceeding a count for 16 ms, then the value is corrected to OFF_H . Before leaving this part of the ISS and returning to the main program, a flag (CTRD) is set to 1 to indicate that the next time a RST 7.5 occurs, the second part of the ISS must be executed.

The second part is entered on the next RST 7.5 interrupt. This time all the five latches are read and the correct values for P_{11} , P_{21} , P_{31} , P_{41} are filled in. Then $P_{13} = P_{11} + P_{12}$ is filled in. Likewise P_{23} , P_{33} , P_{43} are computed and filled in. Next, one of the four eligible candidates P_i (i.e. P_{11} , P_{21} , P_{31} , P_{41}) are selected for comparison. The value $1/8 P_i$ is computed and stored. The selected P_i is compared against the other P_c s. Everytime the condition $P_c - P_i > 1/8 P_i$ is met, then the score in the appropriate counter is updated. In this way all the 4 candidates are compared.

Finally the scores in the respective counters are compared and the majority winner is output as the pitch period.

However if the majority winner fails to make a score of at least 4, then a "hiss" decision is made. Also if any two or more of the candidates are $\emptyset FF_H$, then again a "hiss" decision is made.

In the above implementation one observation made is that if the pitch changes suddenly between two 10 ms segments then a hiss decision is made before the new pitch period is put out. Appropriate interpolation at the receiver is likely to overcome this problem.

CHAPTER 5

SUGGESTIONS FOR FURTHER WORK

5.1 IMPROVEMENTS FOR PRESENT IMPLEMENTATION

The present implementation has been based on available components. However the complete circuit beyond the 2920 ICS may be drastically reduced in component count by using 8253 programmable interval timer/counter, Figure 5.1.

The advantages perceived in implementing the counters on the 8253 may be listed as under:

- a) Drastically reduced component count both in the number of IC's and passive components.
- b) Reduction in the power consumption would be substantial since only two 8253 are sufficient to implement all the four counters as well as have timers to provide the timing for the interrupt control. The 8282 latches each consuming 1 watt power can be dispensed with.
- c) The total number of monostables used at present is eight. Changing over to 8253 provides automatically for the conditions when count exceeds 16 msec. Therefore only four monostables need to be retained for providing the blanking pulses.
- d) The additional logic circuitry used for reset of the XR 2240 is therefore automatically dispensed with.
- e) Higher precision may be achieved since the individual pitch period estimates can be made on 16 bit counters

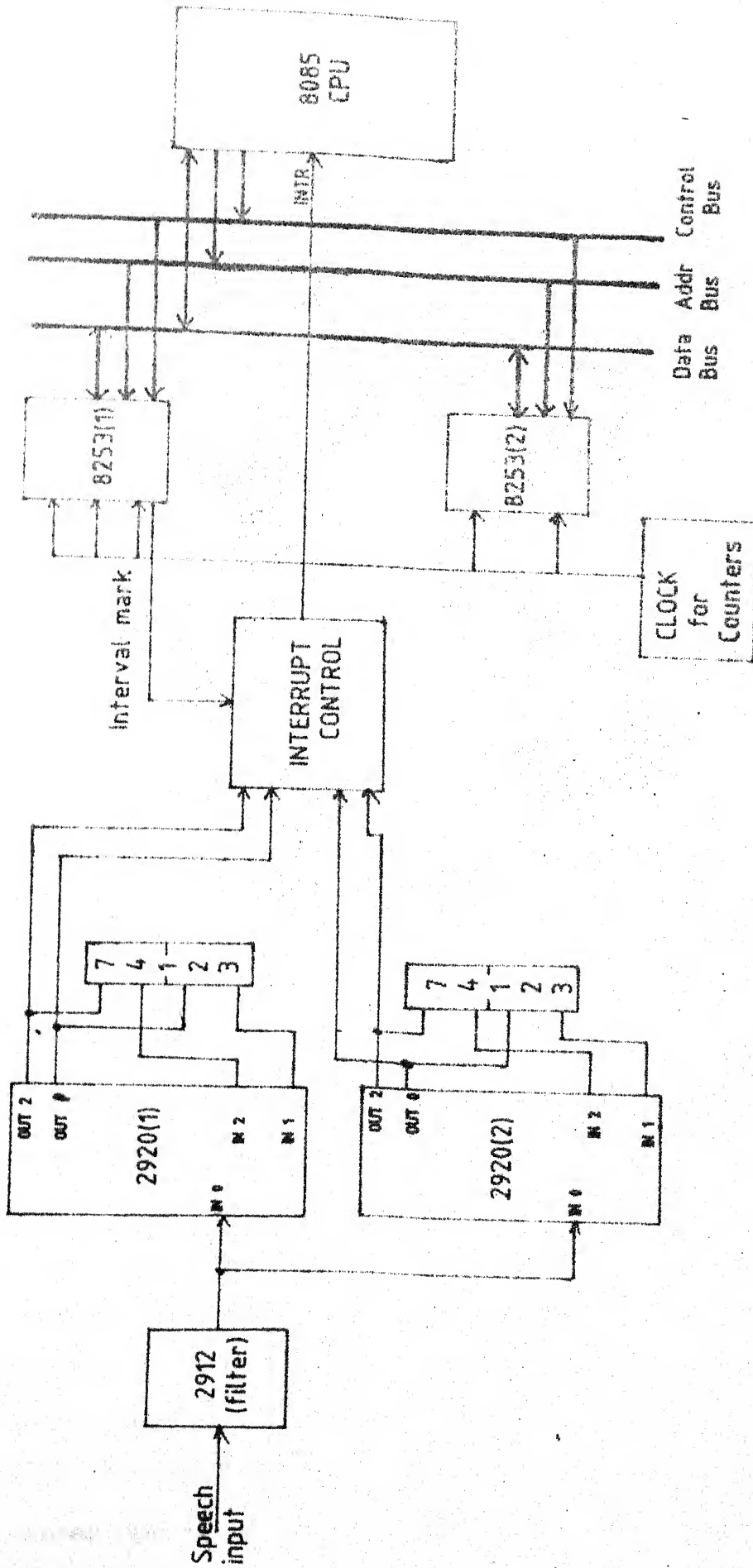


Fig 5.1 Block diagram for suggested improvement

as against 8 bits in the present implementation.

5.2 SUGGESTED SCHEME FOR THE REALISATION OF LPC VOCODER

The pitch period estimator implemented can be used as the excitation analyser for an LPC vocoder based on the schematic of Figure 5.2. Study has shown the feasibility of such a project [19]. The major signal processing hardware required is a combination filter-codec (AMI S3507) and two Signal Processing Interfaces (SPI). One SPI is used to implement the LPC analyzer and the other an acoustic tube synthesizer. The SPI used can be the AMI S2811 or the equivalent NEC PD 7720, both of which are well suited. The 8085 is used for data transfer, control and multiplexing functions along with communications with the host terminal. The overall scheme of implementation corresponds to that shown in Figure 2.12.

5.2.1 Analyzer Implementation

The analyzer is a windowed autocorrelator followed by Durbin's recursion to evaluate the reflection coefficients (Section 2.3). The SPI receives speech samples as serial data from the combo codec-filter, once per analysis frame, on command from the control processor during an A/D interrupt service. The received speech samples are weighted by a Hamming window and the $P+1$ correlation coefficients are computed for the current frame. This computation concludes the interrupt service routine.

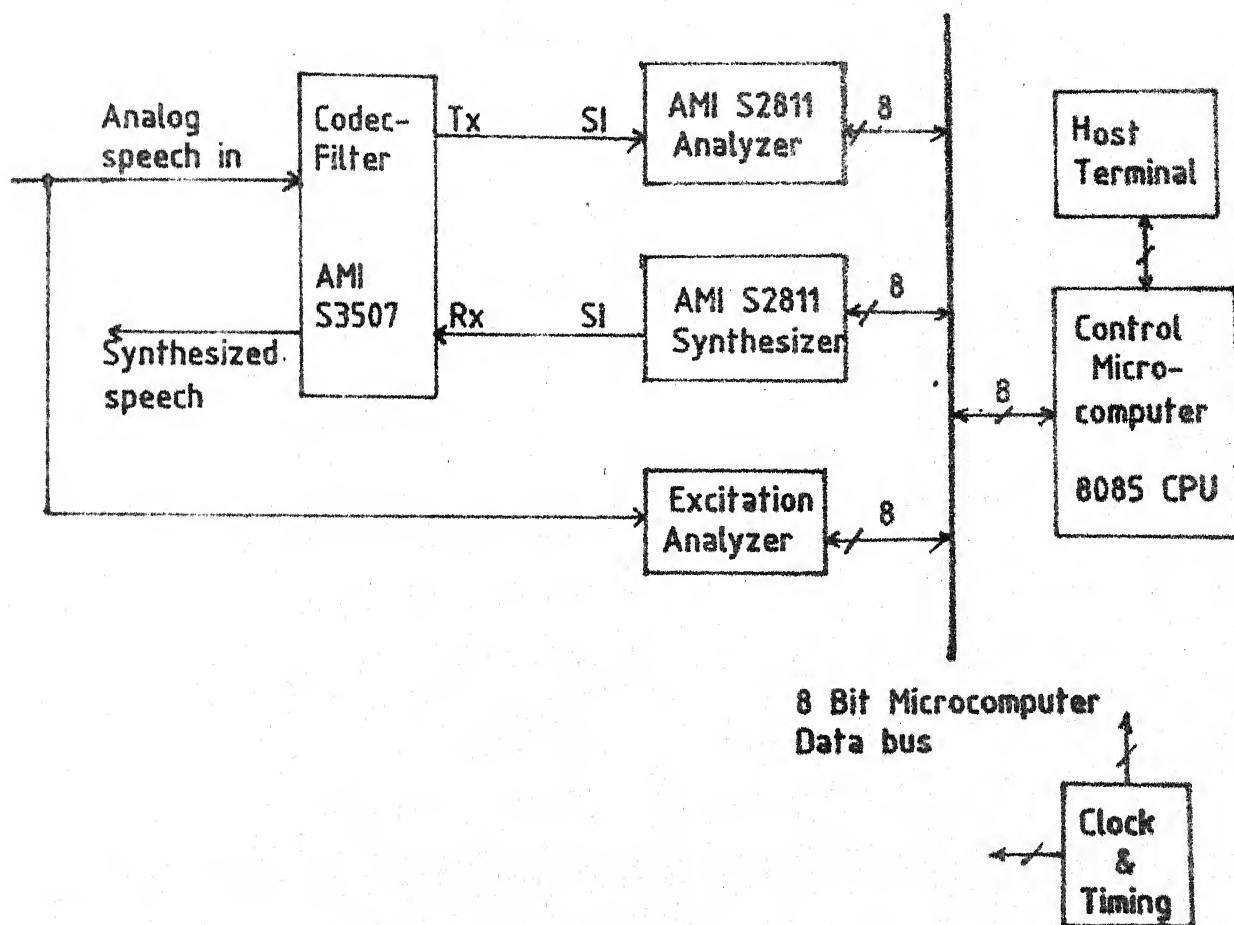


Fig 5.2 LPC Vocoder Architecture

The background routine in the SPI computes the LPC reflection coefficients passed from the interrupt service routine above. This computation, also, is performed once per frame on command from the control processor. The end result of the computation is an array consisting of P reflection coefficients and the prediction residual energy. Parameter coding is implemented in the control processor in order to maintain the flexibility of the SPI analyzer. The analysis frame length is 22.5 ms for 8 KHz sampling rate for an intended predictor order often.

The pitch period estimator already implemented can provide a pitch estimate/unvoiced decision once per the 22.5 ms frame.

5.2.2 Synthesizer Implementation

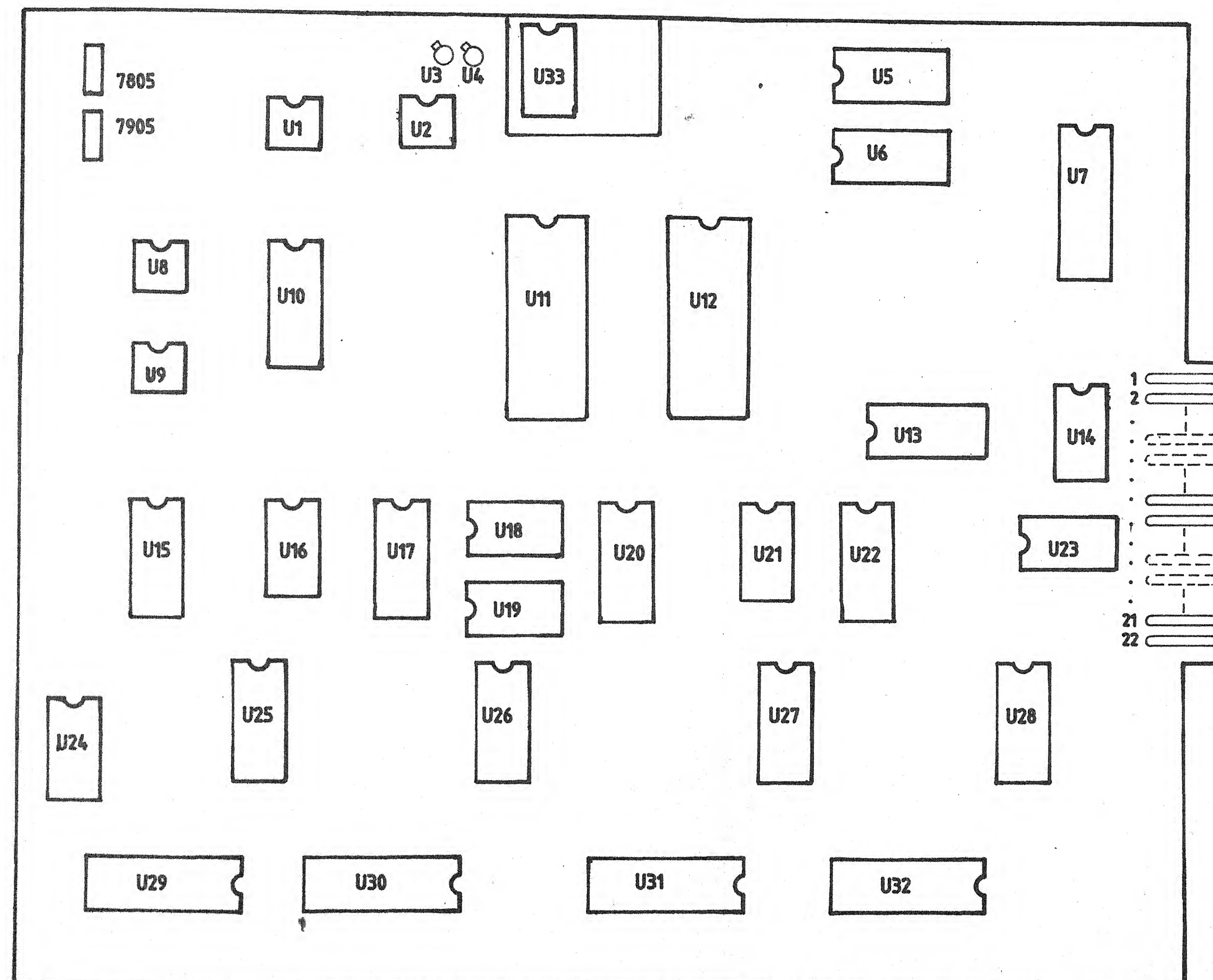
In each frame, the SPI synthesizer receives an energy estimate, pitch/voicing decision, and a set of reflection coefficients from the control and communications microprocessor. The synthesizer reconstructs the speech and outputs it as PCM data through the SPI serial output port. The synthesizer consists of an excitation generator, a lattice (acoustic tube) filter and a one pole de-emphasis filter. The lattice filter coefficients are obtained from a linear interpolation of the past and present frame's reflection coefficients. In voiced frames, the filter excitation is a pulse train with pitch equal to the estimate and amplitude based on a linear interpolation of the

can be drastically reduced as suggested. Lack of a facility to test the estimator for real speech input samples inhibits comments on its performance though tests have otherwise shown it to be fairly accurate in the extraction of pitch period of signals generated using a Laboratory function generator.



REFERENCES

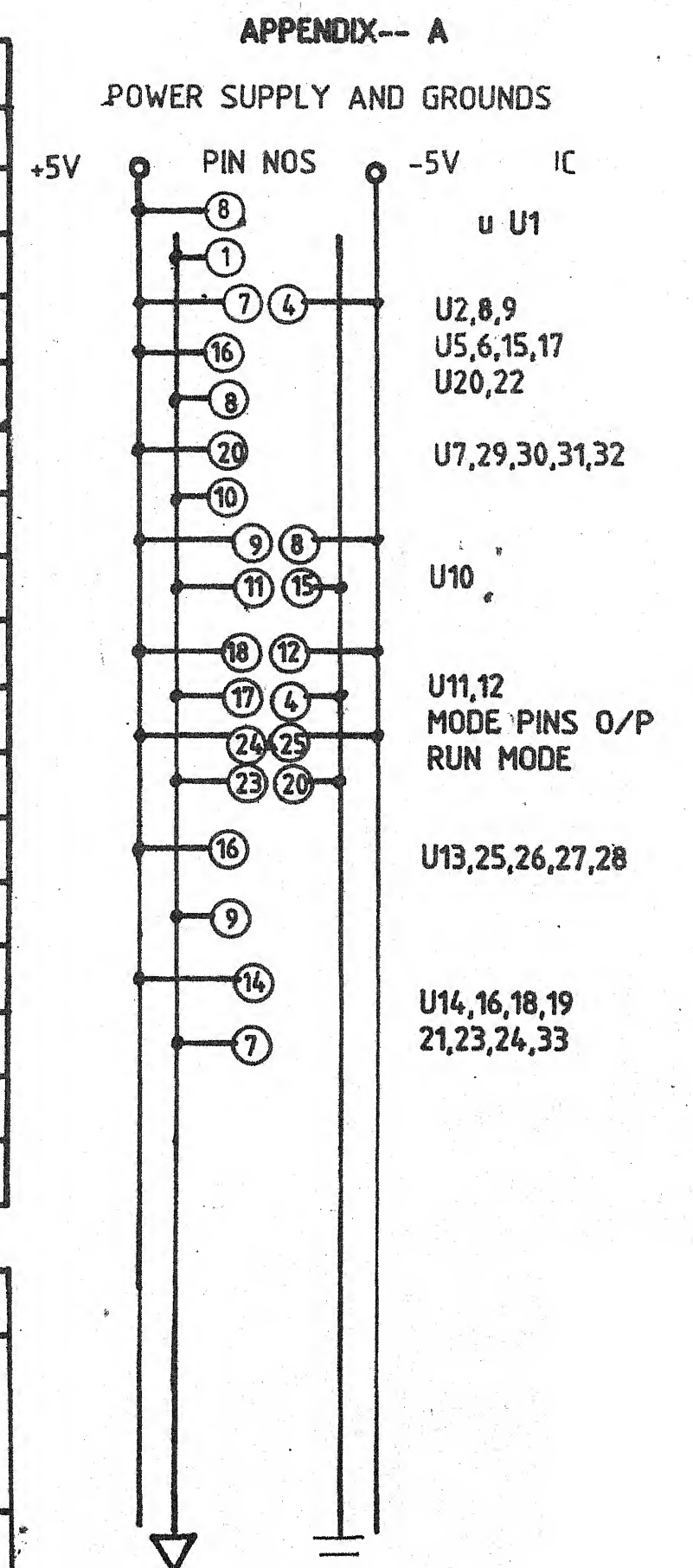
- [1] FLANAGAN J.L., Speech Analysis, Synthesis and Perception, Springer-Verlag.
- [2] JAYANT N.S., 'Digital Coding of Speech Waveforms: PCM, DPCM and DM Quantizers', Proceeding IEEE, May 1974.
- [3] OPPENHEIM A.V., Applications of Digital Signal Processing, Prentice Hall.
- [4] RABINER L.R., SCHAFER R.W., Digital Signal Processing of Speech Signals, Prentice Hall.
- [5] MARKEL J.D., GRAY A.H., Linear Prediction of Speech, Springer Verlag.
- [6] GOLD B., RABINER L.R., 'Parallel Processing Techniques for Estimating Pitch Periods of Speech in the Time Domain', Journal of the Acou. Soc. of America (JASA), August 1969.
- [7] GOLD B., 'Digital Speech Networks', Proc. IEEE, Dec. 1977.
- [8] BEAUCHAMP K., YUEN C., Digital Methods for Signal Analysis, George Allen & Unwin.
- [9] OPPENHEIM A.V., SCHAFER R.W., Digital Signal Processing, Prentice Hall.
- [10] OPPENHEIM A.V., SCHAFER R.W., 'Homomorphic Analysis of Speech', IEEE Trans. on Audio and Electroacoustics, June 1968.
- [11] RABINER L.R., SCHAFER R.W., 'Digital Representation of Speech Signals', Proc. IEEE, April 1975.
- [12] MARKEL J., 'Linear Prediction: A Tutorial Review', Proc. IEEE, April 1975.
- [13] RABINER L.R., et al., 'A Comparative Study of Several Pitch Detection Algorithms', IEEE Trans. on ASSP, Oct. 1976.
- [14] FLANAGAN J.L., 'Automatic Extraction of Formant Frequencies for Continuous Speech', JASA, Jan. 1956.
- [15] ATAL B.S., HANAUER S.L., 'Speech Analysis and Synthesis by Linear Prediction of the Speech Wave', JASA, Aug. 1971.

- [16] DUDLEY H., 'The Vocoder', Bell Laboratories Record, Dec. 1939.
- [17] GOLD B., RADER C.M., 'The Channel Vocoder', IEEE Trans. on Audio and Electroacoustics, Dec. 1967.
- [18] HOFFSTETTER E.M., et al., 'Microprocessor Realisation of a Linear Predictive Vocoder', IEEE Trans. on ASSP, Oct. 1977.
- [19] FELDMAN J.A., et al., 'A Compact Flexible Vocoder Based on a Commercial Signal Processing Microcomputer', IEEE Journal of Solid State Circuits, Feb. 1983.
- [20] Intel Components Catalogue, 1982.
- [21] 2920 Analog Signal Processor Design Handbook.
- [22] 2920 Signal Processing Software:
 - SPAS 20 Compiler Users' Manual
 - 2920 Simulator Users' Manual
 - 2920 Assembly Language Manual.
- [23] LERNER RM, 'Band Pass Filters with Linear Phase', Proc. IEEE, March 1964.

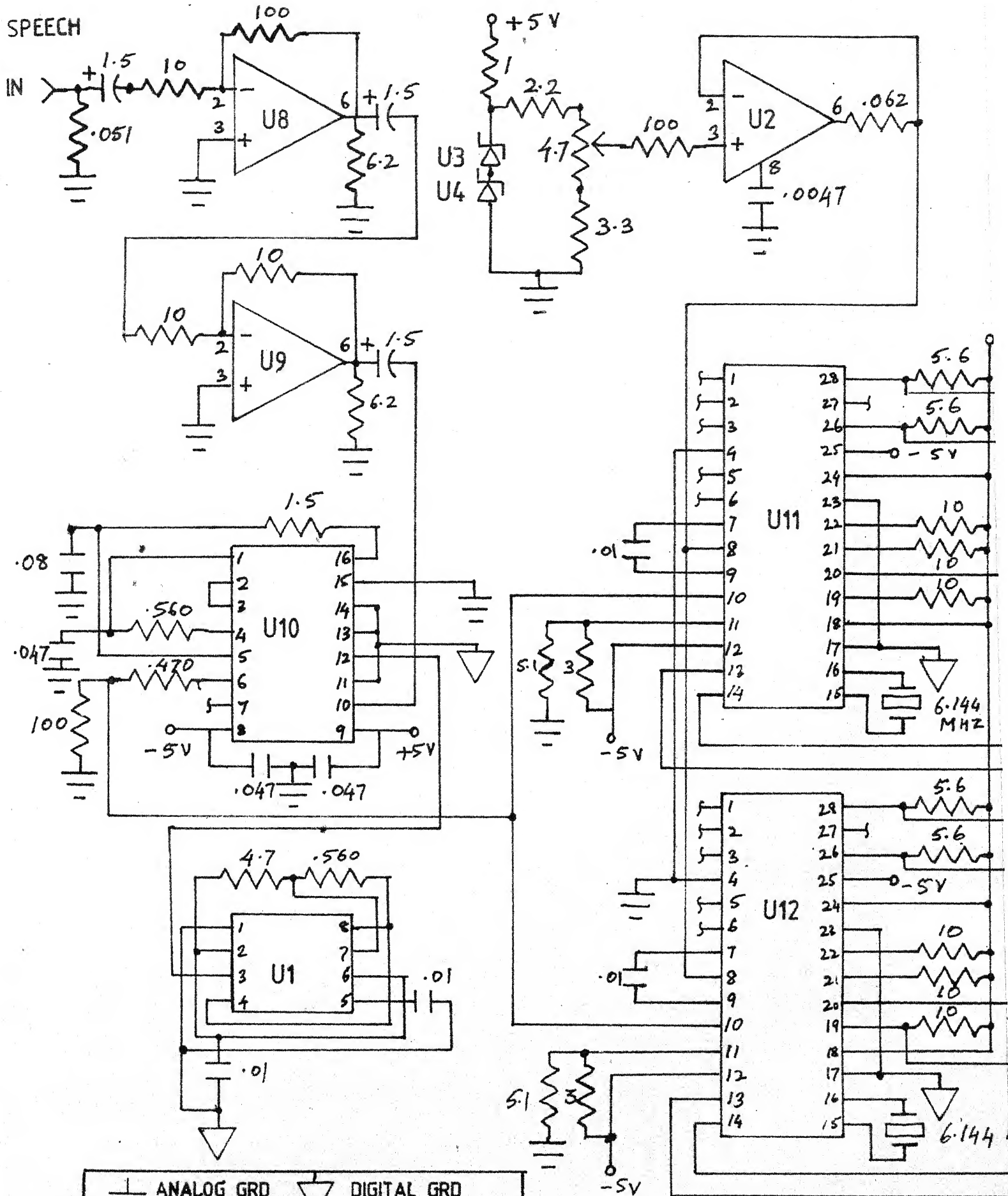


PC CARD LAYOUT

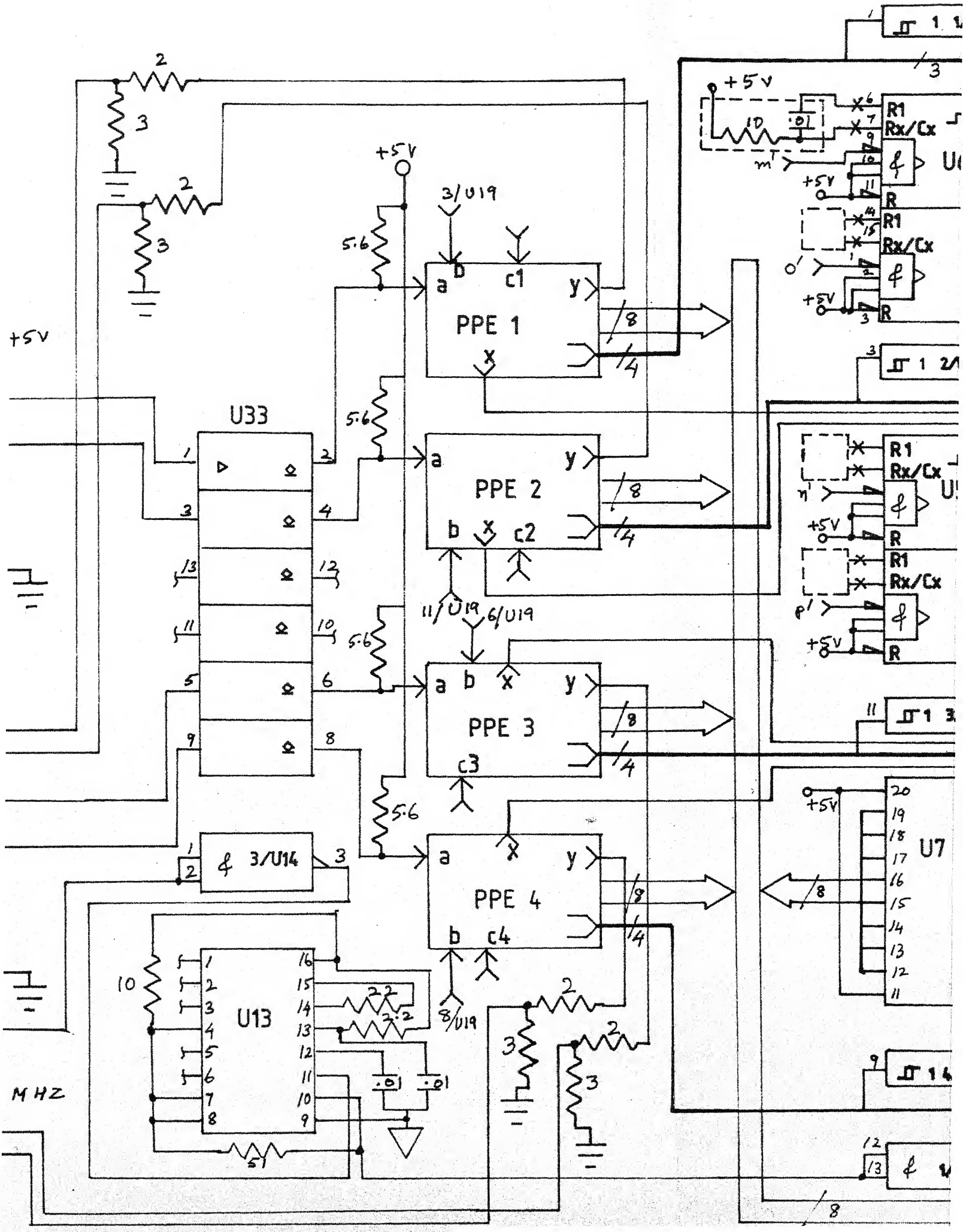
TABLE	
COMPONENT	IC NO.
U1	NES55
U2	LM 308
U3,4	LM 103
U5,6,15,17,20,22	74123
U7,29,30,31,32	8202
U8,9	LM 741
U10	2912
U11,12	2920
U13,25,26,27,28	XR 2240
U14,18	7400
U16,21	74 LS20
U19	74 LS32
U19 u U23	74 LS155
U24	7414
U33	7407
GROUNDS	
	DIGITAL GROUND
	ANALOG GROUND
GROUNDS CONNECTED AT P/S	



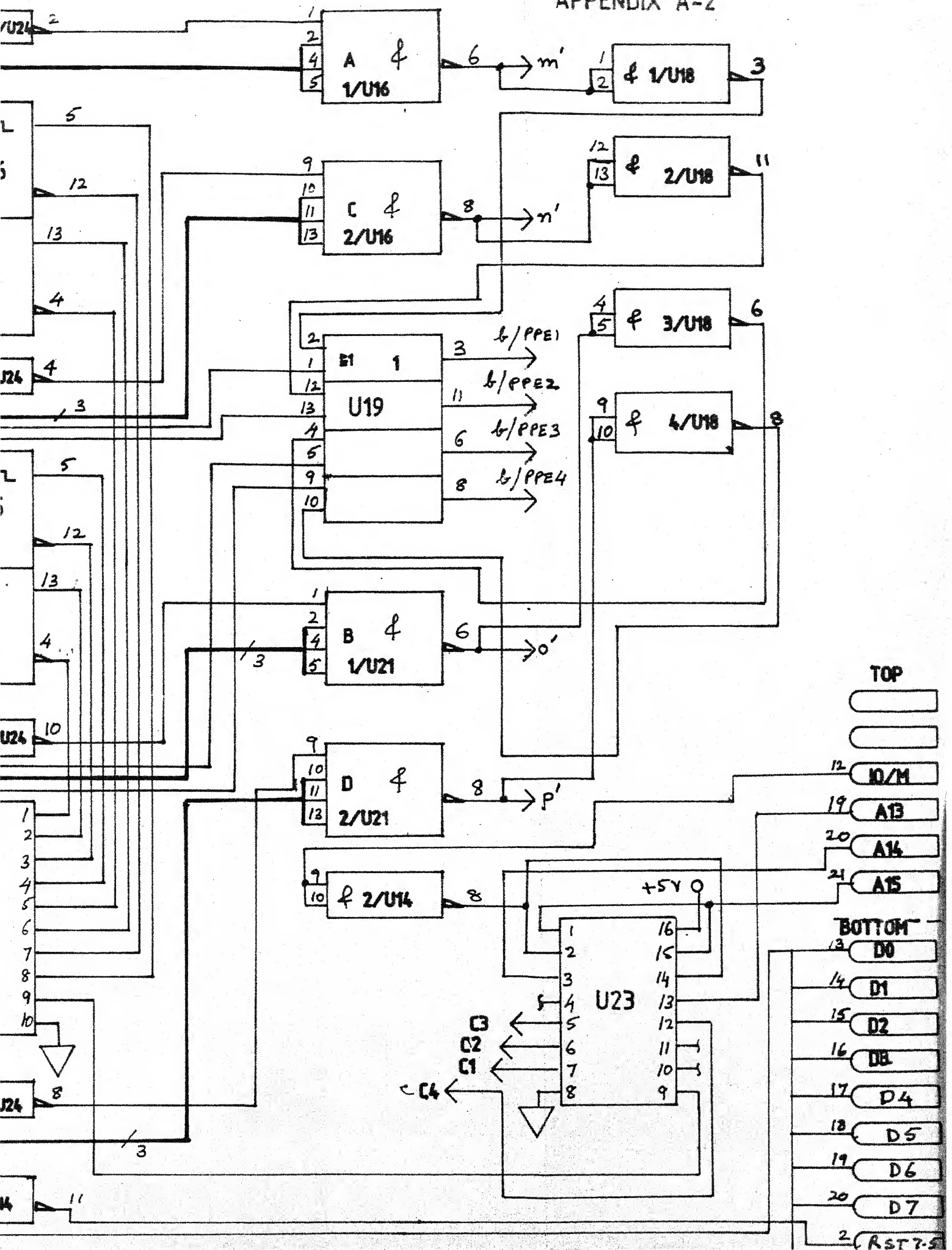
SPEECH



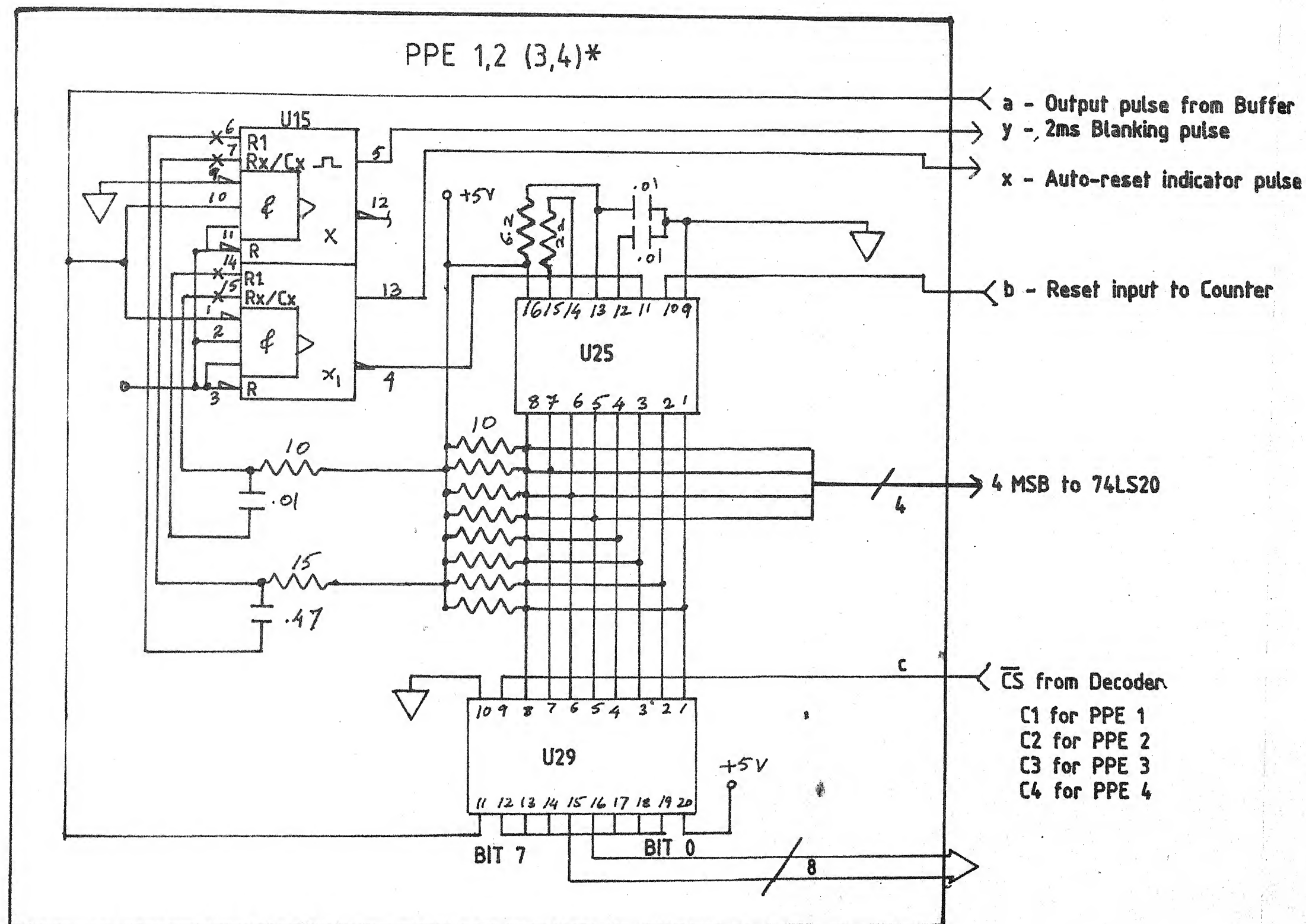
DETAIL CIRCUIT DIAGRAM



APPENDIX A-2



APPENDIX A-3



* IN PPE 3,4 PINS OF X & X1 OF 74123 ARE INTERCHANGED (U20,U22)

CIRCUIT OF A PPE

PPE 1	U15
	U25
	U29
PPE 2	U17
	U26
	U30
PPE 3	U20
	U27
	U31
PPE 4	U22
	U28
	U32

DIGITAL GROUND

ANALOG GROUND

ALL CAP IN MICROFARADS
ALL RESISTORS IN KILO OHMS

ASSEMBLER INVOKED BY: :F1:AS2920 FULFIL

LINE LOC OBJECT SOURCE STATEMENT

```

1
2      ; A/D CONVERSION ROUTINE ADDED BY MACRO ADCONV
3      0 0066EB      SUB DAR, DAR, R00, IN0
4      1 0000EF      IN0
5      2 0000EF      IN0
6      3 0000EF      IN0
7      4 0000EF      IN0
8      5 0000EF      IN0
9      6 0000EF      IN0
10     7 4000EF      NOP
11     8 6000EF      CVTS
12     9 4000EF      NOP
13    10 EBE6ED      ADD DAR, KM2, R00, CND6
14    11 4000EF      NOP
15    12 7100EF      CVT7
16    13 4000EF      NOP
17    14 6100EF      CVT6
18    15 4000EF      NOP
19    16 5100EF      CVT5
20    17 4000EF      NOP
21    18 4100EF      CVT4
22    19 4000EF      NOP
23    20 3100EF      CVT3
24    21 4000EF      NOP
25    22 2100EF      CVT2
26    23 4000EF      NOP
27    24 1100EF      CVT1
28    25 4000EF      NOP
29    26 0100EF      CVT0
30    27 4000EF      NOP
31    28 4022EF      LDA INPUT, DAR      ; SCALE INPUT HERE
32      ; END OF MACRO ADCONV
33      ; THESE ARE THE SCALINGS SUGGESTED BY : CODFIL FOR THE VARIOUS INPUTS:
34    29 4000FE      LDA IN0+P0, INPUT, R08
35    30 4400CE      LDA IN0+P1, INPUT, R07
36      IN0+P2 EQU IN0+P1
37      IN0+P3 EQU IN0+P0
38      OUT2+P0 EQU TEMP
39    31 4C00FF      LDA OUT2+P0, OUT1+P0, R00
40      ; OUT2+P0=1.00000000*OUT1+P0
41    32 4818EF      LDA OUT1+P0, OUT0+P0, R00
42      ; OUT1+P0=1.00000000*OUT0+P0
43    33 42185E      LDA OUT0+P0, OUT2+P0, R03
44      ; OUT0+P0=0.125000000*OUT2+P0
45    34 4218FB      SUB OUT0+P0, OUT2+P0, R00
46      ; OUT0+P0=-0.87500000*OUT2+P0
47    35 42189D      ADD OUT0+P0, OUT2+P0, R13
48      ; OUT0+P0=-0.87487792*OUT2+P0
49    36 42183D      ADD OUT0+P0, OUT2+P0, R10
50      ; OUT0+P0=-0.87390136*OUT2+P0
51    37 4218DC      ADD OUT0+P0, OUT2+P0, R07
52      ; OUT0+P0=-0.86608886*OUT2+P0

```

LINE LOC OBJECT SOURCE STATEMENT

```

53 38 48187D ADD OUT0+P0, OUT0+P0, R12
54      ; OUT0+P0=-0.86630039*OUT2+P0
55 39 4218BA SUB OUT0+P0, OUT2+P0, R06
56      ; OUT0+P0=-0.88192539*OUT2+P0
57 40 4810DD ADD OUT0+P0, OUT1+P0, L01
58      ; OUT0+P0=2.00000000*OUT1+P0-0.88192539*OUT2+P0
59 41 48105A SUB OUT0+P0, OUT1+P0, R03
60      ; OUT0+P0=1.87500000*OUT1+P0-0.88192539*OUT2+P0
61 42 48105B SUB OUT0+P0, OUT1+P0, R11
62      ; OUT0+P0=1.8745117*OUT1+P0-0.88192539*OUT2+P0
63 43 4018FD ADD OUT0+P0, IN0+P0, R00
64      ; OUT0+P0=1.8745117*OUT1+P0-0.88192539*OUT2+P0+1.00000000*IN0+P0
65      OUT2+P1 EQU TEMP
66 44 4E00FF LDA OUT2+P1, OUT1+P1, R00
67      ; OUT2+P1=1.00000000*OUT1+P1
68 45 4E18EF LDA OUT1+P1, OUT0+P1, R00
69      ; OUT1+P1=1.00000000*OUT0+P1
70 46 46185E LDA OUT0+P1, OUT2+P1, R03
71      ; OUT0+P1=0.125000000*OUT2+P1
72 47 4E185B SUB OUT0+P1, OUT0+P1, R11
73      ; OUT0+P1=0.124938964*OUT2+P1
74 48 1066EB SUB DAR, DAR, R00, IN1
75 49 1E189B SUB OUT0+P1, OUT0+P1, R13, IN1
76      ; OUT0+P1=0.124923718*OUT2+P1
77 50 1618FB SUB OUT0+P1, OUT2+P1, R00, IN1
78      ; OUT0+P1=-0.87507626*OUT2+P1
79 51 1E18DC ADD OUT0+P1, OUT0+P1, R07, IN1
80      ; OUT0+P1=-0.88191279*OUT2+P1
81 52 1E10DD ADD OUT0+P1, OUT1+P1, L01, IN1
82      ; OUT0+P1=2.00000000*OUT1+P1-0.88191279*OUT2+P1
83 53 1E105A SUB OUT0+P1, OUT1+P1, R03, IN1
84      ; OUT0+P1=1.87500000*OUT1+P1-0.88191279*OUT2+P1
85 54 4E10FA SUB OUT0+P1, OUT1+P1, R08
86      ; OUT0+P1=1.8710937*OUT1+P1-0.88191279*OUT2+P1
87 55 6E103B SUB OUT0+P1, OUT1+P1, R10, CVTS
88      ; OUT0+P1=1.8701171*OUT1+P1-0.88191279*OUT2+P1
89 56 4E107B SUB OUT0+P1, OUT1+P1, R12
90      ; OUT0+P1=1.8698730*OUT1+P1-0.88191279*OUT2+P1
91 57 EBE6ED ADD DAR, KM2, R00, CND6
92 58 4610FD ADD OUT0+P1, IN0+P1, R00
93      ; OUT0+P1=1.8698730*OUT1+P1-0.88191279*OUT2+P1+1.00000000*IN0+P1
94 59 7100EF CVT7
95      OUT2+P2 EQU TEMP
96      ; INCLUDE TWO INST TO TEST BITA:
97 60 F9CACF LDA BITA, KP5, L01, CND7
98 61 70C2EF LDA BITA, KP0, CND5
99      ; CONTINUE FILTERING
100 62 4428FF LDA OUT2+P2, OUT1+P2, R00
101      ; OUT2+P2=1.00000000*OUT1+P2
102 63 4260FF LDA OUT1+P2, OUT0+P2, R00
103      ; OUT1+P2=1.00000000*OUT0+P2
104 64 46484E LDA OUT0+P2, OUT2+P2, R03
105      ; OUT0+P2=0.125000000*OUT2+P2
106 65 4648EB SUB OUT0+P2, OUT2+P2, R00

```

LINE LOC OBJECT SOURCE STATEMENT

```

107      ; OUT0+P2=-0.87500000*OUT2+P2
108  66 4660CC ADD OUT0+P2,OUT0+P2,R07
109      ; OUT0+P2=-0.88183593*OUT2+P2
110  67 46606D ADD OUT0+P2,OUT0+P2,R12
111      ; OUT0+P2=-0.88205126*OUT2+P2
112  68 46488D ADD OUT0+P2,OUT2+P2,R13
113      ; OUT0+P2=-0.88192919*OUT2+P2
114  69 4468CD ADD OUT0+P2,OUT1+P2,L01
115      ; OUT0+P2=2.00000000*OUT1+P2-0.88192919*OUT2+P2
116  70 44684A SUB OUT0+P2,OUT1+P2,R03
117      ; OUT0+P2=1.87500000*OUT1+P2-0.88192919*OUT2+P2
118  71 4468AA SUB OUT0+P2,OUT1+P2,R06
119      ; OUT0+P2=1.85937500*OUT1+P2-0.88192919*OUT2+P2
120  72 4468EA SUB OUT0+P2,OUT1+P2,R08
121      ; OUT0+P2=1.8554687*OUT1+P2-0.88192919*OUT2+P2
122  73 44684B SUB OUT0+P2,OUT1+P2,R11
123      ; OUT0+P2=1.8549804*OUT1+P2-0.88192919*OUT2+P2
124  74 44688D ADD OUT0+P2,OUT1+P2,R13
125      ; OUT0+P2=1.8551025*OUT1+P2-0.88192919*OUT2+P2
126  75 4640ED ADD OUT0+P2,IN0+P2,R00
127      ; OUT0+P2=1.8551025*OUT1+P2-0.88192919*OUT2+P2+1.00000000*IN0+P2
128      OUT2+P3 EQU TEMP
129  76 3066EB SUB DAR,DAR,R00,IN3      ; A DUMMY READ TO DISCHARGE S&H CAPACITOR
130  77 3628FF LDA OUT2+P3,OUT1+P3,R00,IN3
131      ; OUT2+P3=1.00000000*OUT1+P3
132  78 3C60FF LDA OUT1+P3,OUT0+P3,R00,IN3
133      ; OUT1+P3=1.00000000*OUT0+P3
134  79 32584E LDA OUT0+P3,OUT2+P3,R03,IN3
135      ; OUT0+P3=0.125000000*OUT2+P3
136  80 3258EB SUB OUT0+P3,OUT2+P3,R00,IN3
137      ; OUT0+P3=-0.87500000*OUT2+P3
138  81 3870AC ADD OUT0+P3,OUT0+P3,R06,IN3
139      ; OUT0+P3=-0.88867187*OUT2+P3
140  82 48702B SUB OUT0+P3,OUT0+P3,R10
141      ; OUT0+P3=-0.88780400*OUT2+P3
142  83 2066EB SUB DAR,DAR,R00,IN2      ; START ACQUIRING BITS
143  84 2258CC ADD OUT0+P3,OUT2+P3,R07,IN2
144      ; OUT0+P3=-0.87999150*OUT2+P3
145  85 22580B SUB OUT0+P3,OUT2+P3,R09,IN2
146      ; OUT0+P3=-0.88194462*OUT2+P3
147  86 2278CD ADD OUT0+P3,OUT1+P3,L01,IN2
148      ; OUT0+P3=2.00000000*OUT1+P3-0.88194462*OUT2+P3
149  87 22784A SUB OUT0+P3,OUT1+P3,R03,IN2
150      ; OUT0+P3=1.87500000*OUT1+P3-0.88194462*OUT2+P3
151  88 22788A SUB OUT0+P3,OUT1+P3,R05,IN2
152      ; OUT0+P3=1.84375000*OUT1+P3-0.88194462*OUT2+P3
153  89 42782D ADD OUT0+P3,OUT1+P3,R10
154      ; OUT0+P3=1.8447265*OUT1+P3-0.88194462*OUT2+P3
155  90 62786D ADD OUT0+P3,OUT1+P3,R12,CVTS
156      ; OUT0+P3=1.8449707*OUT1+P3-0.88194462*OUT2+P3
157  91 4058ED ADD OUT0+P3,IN0+P3,R00
158      ; OUT0+P3=1.8449707*OUT1+P3-0.88194462*OUT2+P3+1.00000000*IN0+P3
159  92 EBE6ED ADD DAR,KM2,CND6      ; A/D CONV INST
160      ; NOW THE SUM OF THE SUB FILTER OUTPUTS HAS TO BE TAKEN

```

LINE LOC OBJECT SOURCE STATEMENT

```
161 93 4858FF LDA OUTPUT, OUT0+P0, R00
162 94 7B58FB SUB OUTPUT, OUT0+P1, R00, CVT7
163 95 4270FD ADD OUTPUT, OUT0+P2, R00
164 ; INCLUDE TWO INST TO SET BIT B
165 96 FDDACF LDA BITB, KP5, L01, CND7
166 97 74D2EF LDA BITB, KP0, CNDS
167 ; CONTINUE
168 98 4870FB SUB OUTPUT, OUT0+P3, R00
169 99 4878DD ADD OUTPUT, OUTPUT, L01
170 100 4878DF LDA OUTPUT, OUTPUT, L01
171 101 486CEF LDA DAR, OUTPUT
172 102 4000EF NOP
173 103 4000EF NOP
174 104 4000EF NOP
175 105 44D2FF LDA NEGVAL, KP0 ; PEAK DETECTION AND RUN DOWN SECTION
176 106 7CDADF LDA NEGVAL, KP5, L01, CNDS
177 107 4C28F7 ABS TEMP, OUTPUT
178 108 448A1A SUB TEMP, KP1, R01 ; TO SET A THRESHOLD FOR INPUT SIGNAL LEVEL
179 109 424CE5 LIM DAR, TEMP
180 110 4000EF NOP
181 111 70D2FF LDA OUTPUT, KP0, CNDS
182 112 4829E7 ABS CEE, OUTPUT
183 113 4281FF LDA SAVE, BEE
184 114 4689EB SUB BEE, AYE
185 115 42C4E5 LIM DAR, BEE
186 116 7093EF LDA FLAG, KP0, CNDS
187 117 F393EF LDA FLAG, KP2, CND7
188 118 4489EF LDA BEE, SAVE
189 119 4481EB SUB BEE, CEE
190 120 42C4E5 LIM DAR, BEE
191 121 F393ED ADD FLAG, KP2, CND7
192 122 4093FF LDA PEAK, KP0
193 123 48C4EF LDA DAR, FLAG
194 124 4000EF NOP
195 125 4000EF NOP
196 126 F199FF LDA PEAK, SAVE, CND7
197 127 4481EF LDA BEE, CEE
198 128 4489FF LDA AYE, SAVE
199 129 4A6CEF LDA DAR, NEGVAL
200 130 4493EF LDA NPEAK, KP0
201 131 4C99FF LDA PPEAK, PEAK
202 132 F593FF LDA PPEAK, KP0, CND7
203 133 FD99EF LDA NPEAK, PEAK, CND7
204 134 4064EF LDA DAR, BITA
205 135 4000EF NOP
206 136 F593FF LDA PPEAK, KP0, CND7
207 137 4A64EF LDA DAR, BITB
208 138 4000EF NOP
209 139 F593EF LDA NPEAK, KP0, CND7
210 140 40E9EF LDA PSTORE, PCRDV
211 141 4AC9FB SUB PCRDV, PPEAK
212 142 40ECE5 LIM DAR, PCRDV
213 143 4EEBEF LDA PINFO, KM1
214 144 7EC9FF LDA POSPK, PPEAK, CNDS
```


LINE LOC OBJECT SOURCE STATEMENT

```

215 145 7CCBCF LDA PINFO, KP5, L01, CNDS
216 146 42E4E5 LIM DAR, PINFO
217 147 40E1FF LDA PCRDV, PSTORE
218 148 F3E9FF LDA PCRDV, POSPK, CND7
219 149 8000EF OUT0
220 150 8000EF OUT0
221 151 8000EF OUT0
222 152 8000EF OUT0
223 153 8000EF OUT0
224 154 8000EF OUT0
225 155 48F9EF LDA NSTORE, NCRDV
226 156 4AD1FB SUB NCRDV, NPEAK
227 157 48ECE5 LIM DAR, NCRDV
228 158 4EFBEF LDA NINFO, KM1
229 159 7ED1FF LDA NEGPK, NPEAK, CNDS
230 160 7CDBC F LDA NINFO, KP5, L01, CNDS
231 161 4AE4E5 LIM DAR, NINFO
232 162 48F1FF LDA NCRDV, NSTORE
233 163 FBF9FF LDA NCRDV, NEGPK, CND7
234 164 A000EF OUT2
235 165 A000EF OUT2
236 166 A000EF OUT2
237 167 A000EF OUT2
238 168 A000EF OUT2
239 169 A000EF OUT2
240 170 4064EF LDA DAR, BITA
241 171 4000EF NOP
242 172 F3E9FF LDA PCRDV, POSPK, CND7
243 173 4A64EF LDA DAR, BITB
244 174 40ACEF LDA PY, PCRDV
245 175 FBF9FF LDA NCRDV, NEGPK, CND7
246 176 40ACAA SUB PY, PCRDV, R06
247 177 40AC0B SUB PY, PCRDV, R09
248 178 40064D ADD PY, PY, R11
249      ; PY=. 9829106*PCRDV-----THE ACTUAL FACTOR DESIRED WAS 0.9828206
250 179 4043FF LDA PCRDV, PY      ; PCRDV HAS LATEST RUNDOWN VALUE
251 180 48ACFF LDA NY, NCRDV
252 181 48ACBA SUB NY, NCRDV, R06
253 182 48AC1B SUB NY, NCRDV, R09
254 183 400E5D ADD NY, NY, R11
255      ; NY=. 98291056*NCRDV-----THE SAME ERROR EXSITS
256 184 405BFF LDA NCRDV, NY      ; NCRDV HAS NOW THE LATEST RUNDOWN VALUE FOR NEGATIVE PEAKS
257 185 4000EF NOP
258 186 4000EF NOP
259 187 4000EF NOP
260 188 5000EF EDP
261 189 4000EF NOP
262 190 4000EF NOP
263 191 4000EF NOP
264      END

```

SYMBOL:

VALUE:

INPUT	0
INO+P0	1
INO+P1	2
INO+P2	2
INO+P3	1
OUT2+P0	3
TEMP	3
OUT1+P0	4
OUT0+P0	5
OUT2+P1	3
OUT1+P1	6
OUT0+P1	7
OUT2+P2	3
BITA	8
OUT1+P2	9
OUT0+P2	10
OUT2+P3	3
OUT1+P3	11
OUT0+P3	12
OUTPUT	13
BITB	14
NEGVAL	15
CEE	16
SAVE	17
BEE	18
AYE	19
FLAG	20
PEAK	21
NPEAK	22
PPEAK	23
PSTORE	24
PCRDV	25
PINFO	26
POSPK	27
NSTORE	28
NCRDV	29
NINFO	30
NEGPK	31
PY	32
NY	33

ASSEMBLY COMPLETE

ERRORS	=	0
WARNINGS	=	0
RAMSIZE	=	34
ROMSIZE	=	192

ASSEMBLER INVOKED BY: :F1:AS2920 FINFIL

LINE LOC OBJECT SOURCE STATEMENT

```
1
2      ; A/D CONVERSION ROUTINE ADDED BY MACRO ADCONV
3      0 0066EB      SUB DAR, DAR, R00, IN0
4      1 0000EF      IN0
5      2 0000EF      IN0
6      3 0000EF      IN0
7      4 0000EF      IN0
8      5 0000EF      IN0
9      6 0000EF      IN0
10     7 4000EF      NOP
11     8 6000EF      CVT5
12     9 EBE6ED      ADD DAR, KM2, R00, CND6
13    10 4000EF      NOP
14    11 7100EF      CVT7
15    12 4000EF      NOP
16    13 6100EF      CVT6
17      OUT2+P0 EQU TEMP
18    14 4008EF LDA OUT2+P0, OUT1+P0, R00, NOP
19      ; OUT2+P0=1.00000000*OUT1+P0
20    15 5300FF LDA OUT1+P0, OUT0+P0, R00, CVT5
21      ; OUT1+P0=1.00000000*OUT0+P0
22    16 44004E LDA OUT0+P0, OUT2+P0, R03, NOP
23      ; OUT0+P0=0.12500000*OUT2+P0
24    17 4500EB SUB OUT0+P0, OUT2+P0, R00, CVT4
25      ; OUT0+P0=-0.87500000*OUT2+P0
26    18 46008A SUB OUT0+P0, OUT0+P0, R05, NOP
27      ; OUT0+P0=-0.84765625*OUT2+P0
28    19 37002B SUB OUT0+P0, OUT0+P0, R10, CVT3
29      ; OUT0+P0=-0.84682841*OUT2+P0
30    20 4400CA SUB OUT0+P0, OUT2+P0, R07, NOP
31      ; OUT0+P0=-0.85464091*OUT2+P0
32    21 2508CD ADD OUT0+P0, OUT1+P0, L01, CVT2
33      ; OUT0+P0=2.00000000*OUT1+P0-0.85464091*OUT2+P0
34    22 44084A SUB OUT0+P0, OUT1+P0, R03, NOP
35      ; OUT0+P0=1.87500000*OUT1+P0-0.85464091*OUT2+P0
36    23 15086A SUB OUT0+P0, OUT1+P0, R04, CVT1
37      ; OUT0+P0=1.81250000*OUT1+P0-0.85464091*OUT2+P0
38    24 4408AC ADD OUT0+P0, OUT1+P0, R06, NOP
39      ; OUT0+P0=1.82812500*OUT1+P0-0.85464091*OUT2+P0
40    25 05080B SUB OUT0+P0, OUT1+P0, R09, CVT0
41      ; OUT0+P0=1.8261718*OUT1+P0-0.85464091*OUT2+P0
42    26 4000EF NOP
43    27 4422FF LDA INPUT, DAR
44    28 4218CE LDA IN0+P0, INPUT, R07      ; INPUT SCALING IS EFECTED HERE
45    29 4218BE LDA IN0+P1, INPUT, R06
46      IN0+P2 EQU IN0+P1
47      IN0+P3 EQU IN0+P1
48      IN0+P4 EQU IN0+P1
49      IN0+P5 EQU IN0+P1
50    30 4C00ED ADD OUT0+P0, IN0+P0, R00
51      ; OUT0+P0=1.8261718*OUT1+P0-0.85464091*OUT2+P0+1.00000000*IN0+P0
52      OUT2+P1 EQU TEMP
```

.0000000

00000000

LINE LOC OBJECT SOURCE STATEMENT

```

53 31 4A00EF LDA OUT2+P1,OUT1+P1,R00
54      ; OUT2+P1=1.00000000*OUT1+P1
55 32 4E18EF LDA OUT1+P1,OUT0+P1,R00
56      ; OUT1+P1=1.00000000*OUT0+P1
57 33 4E107E LDA OUT0+P1,OUT1+P1,R04
58      ; OUT0+P1=0.06250000*OUT1+P1
59 34 4E103A SUB OUT0+P1,OUT1+P1,R02
60      ; OUT0+P1=-0.18750000*OUT1+P1
61 35 4E10DD ADD OUT0+P1,OUT1+P1,L01
62      ; OUT0+P1=1.81250000*OUT1+P1
63 36 4E185D ADD OUT0+P1,OUT0+P1,R11
64      ; OUT0+P1=1.8133850*OUT1+P1
65 37 4410FB SUB OUT0+P1,OUT2+P1,R00
66      ; OUT0+P1=1.8133850*OUT1+P1-1.00000000*OUT2+P1
67 38 44105C ADD OUT0+P1,OUT2+P1,R03
68      ; OUT0+P1=1.8133850*OUT1+P1-0.87500000*OUT2+P1
69 39 4410BC ADD OUT0+P1,OUT2+P1,R06
70      ; OUT0+P1=1.8133850*OUT1+P1-0.85937500*OUT2+P1
71 40 4410FC ADD OUT0+P1,OUT2+P1,R08
72      ; OUT0+P1=1.8133850*OUT1+P1-0.85546875*OUT2+P1
73 41 44103D ADD OUT0+P1,OUT2+P1,R10
74      ; OUT0+P1=1.8133850*OUT1+P1-0.85449218*OUT2+P1
75 42 44109B SUB OUT0+P1,OUT2+P1,R13
76      ; OUT0+P1=1.8133850*OUT1+P1-0.85461425*OUT2+P1
77 43 4C18FD ADD OUT0+P1,IN0+P1,R00
78 44 1066EB SUB DAR,DAR,IN1      ; START ACQUIRING BIT A
79      ; OUT0+P1=1.8133850*OUT1+P1-0.85461425*OUT2+P1+1.00000000*IN0+P1
80      OUT2+P2 EQU TEMP
81 45 1020EF LDA OUT2+P2,OUT1+P2,R00,IN1
82      ; OUT2+P2=1.00000000*OUT1+P2
83 46 1068EF LDA OUT1+P2,OUT0+P2,R00,IN1
84      ; OUT1+P2=1.00000000*OUT0+P2
85 47 10405E LDA OUT0+P2,OUT2+P2,R03,IN1
86      ; OUT0+P2=0.12500000*OUT2+P2
87 48 1040FB SUB OUT0+P2,OUT2+P2,R00,IN1
88      ; OUT0+P2=-0.87500000*OUT2+P2
89 49 1068DA SUB OUT0+P2,OUT0+P2,R07,IN1
90      ; OUT0+P2=-0.86816406*OUT2+P2
91 50 1068DA SUB OUT0+P2,OUT0+P2,R07,IN1
92      ; OUT0+P2=-0.86138154*OUT2+P2
93 51 4068DA SUB OUT0+P2,OUT0+P2,R07,NOP
94      ; OUT0+P2=-0.85465195*OUT2+P2
95 52 6060DD ADD OUT0+P2,OUT1+P2,L01,CVTS
96      ; OUT0+P2=2.00000000*OUT1+P2-0.85465195*OUT2+P2
97 53 40603A SUB OUT0+P2,OUT1+P2,R02
98      ; OUT0+P2=1.75000000*OUT1+P2-0.85465195*OUT2+P2
99 54 EBE6ED ADD DAR,KM2,R00,CND6      ; A/D CONV INST
100 55 40609C ADD OUT0+P2,OUT1+P2,R05,NOP
101      ; OUT0+P2=1.78125000*OUT1+P2-0.85465195*OUT2+P2
102 56 71601B SUB OUT0+P2,OUT1+P2,R09,CVT7
103      ; OUT0+P2=1.77929680*OUT1+P2-0.85465195*OUT2+P2
104 57 40607D ADD OUT0+P2,OUT1+P2,R12
105      ; OUT0+P2=1.77954100*OUT1+P2-0.85465195*OUT2+P2
106      ; INCLUDE TWO INST TO SET BIT A

```

LINE LOC OBJECT SOURCE STATEMENT

```
107 58 FDCACF LDA BITA, KP5, L01, CND7
108 59 74C2EF LDA BITA, KP0, R00, CND5
109 ; CONTINUE FILTERING
110 60 4848FD ADD OUT0+P2, IN0+P2, R00
111 ; OUT0+P2=1.7795410*OUT1+P2-0.85465195*OUT2+P2+1.00000000*IN0+P2
112 OUT2+P3 EQU TEMP
113 61 3228EF LDA OUT2+P3, OUT1+P3, R00, IN3
114 ; OUT2+P3=1.00000000*OUT1+P3
115 62 3C60FF LDA OUT1+P3, OUT0+P3, R00, IN3
116 ; OUT1+P3=1.00000000*OUT0+P3
117 63 30504E LDA OUT0+P3, OUT2+P3, R03, IN3
118 ; OUT0+P3=0.125000000*OUT2+P3
119 64 3050EB SUB OUT0+P3, OUT2+P3, R00, IN3
120 ; OUT0+P3=-0.875000000*OUT2+P3
121 65 3870AA SUB OUT0+P3, OUT0+P3, R06, IN3
122 ; OUT0+P3=-0.86132812*OUT2+P3
123 66 30502D ADD OUT0+P3, OUT2+P3, R10, IN3
124 ; OUT0+P3=-0.86035156*OUT2+P3
125 67 4870CA SUB OUT0+P3, OUT0+P3, R07
126 ; OUT0+P3=-0.85363007*OUT2+P3
127 68 40502B SUB OUT0+P3, OUT2+P3, R10
128 ; OUT0+P3=-0.85460664*OUT2+P3
129 69 4278CD ADD OUT0+P3, OUT1+P3, L01
130 ; OUT0+P3=2.00000000*OUT1+P3-0.85460664*OUT2+P3
131 70 42782A SUB OUT0+P3, OUT1+P3, R02
132 ; OUT0+P3=1.75000000*OUT1+P3-0.85460664*OUT2+P3
133 71 4278AA SUB OUT0+P3, OUT1+P3, R06
134 ; OUT0+P3=1.73437500*OUT1+P3-0.85460664*OUT2+P3
135 72 42786D ADD OUT0+P3, OUT1+P3, R12
136 ; OUT0+P3=1.7346191*OUT1+P3-0.85460664*OUT2+P3
137 73 4858ED ADD OUT0+P3, IN0+P3, R00
138 ; OUT0+P3=1.7346191*OUT1+P3-0.85460664*OUT2+P3+1.00000000*IN0+P3
139 OUT2+P4 EQU TEMP
140 74 2066EB SUB DAR, DAR, R00, IN2 ; START ACQUIRING BIT B
141 75 2828EF LDA OUT2+P4, OUT1+P4, R00, IN2
142 ; OUT2+P4=1.00000000*OUT1+P4
143 76 2A70FF LDA OUT1+P4, OUT0+P4, R00, IN2
144 ; OUT1+P4=1.00000000*OUT0+P4
145 77 2C780C ADD OUT0+P4, OUT1+P4, R01, IN2
146 ; OUT0+P4=1.00000000*OUT0+P4+0.50000000*OUT1+P4
147 78 2C784C ADD OUT0+P4, OUT1+P4, R03, IN2
148 ; OUT0+P4=1.00000000*OUT0+P4+0.62500000*OUT1+P4
149 79 2E708C ADD OUT0+P4, OUT0+P4, R05, IN2
150 ; OUT0+P4=1.03125000*OUT0+P4+0.64453125*OUT1+P4
151 80 4E700D ADD OUT0+P4, OUT0+P4, R09
152 ; OUT0+P4=1.03326416*OUT0+P4+0.64579008*OUT1+P4
153 81 6450EB SUB OUT0+P4, OUT2+P4, R00, CVTS
154 ; OUT0+P4=1.03326416*OUT0+P4+0.64579008*OUT1+P4-1.00000000*OUT2+P4
155 82 44504C ADD OUT0+P4, OUT2+P4, R03
156 ; OUT0+P4=1.03326416*OUT0+P4+0.64579008*OUT1+P4-0.87500000*OUT2+P4
157 83 EBE6ED ADD DAR, KM2, CND6 ; A/D CONV INST
158 84 4450AC ADD OUT0+P4, OUT2+P4, R06
159 ; OUT0+P4=1.03326416*OUT0+P4+0.64579008*OUT1+P4-0.85937500*OUT2+P4
160 85 7550EC ADD OUT0+P4, OUT2+P4, R08, CVT7
```

00000000

00000000

LINE LOC OBJECT SOURCE STATEMENT

```

161      ; OUT0+P4=1.03326416*OUT0+P4+0.64579008*OUT1+P4-0.85546875*OUT2+P4
162 86 44502D ADD OUT0+P4,OUT2+P4,R10
163      ; OUT0+P4=1.03326416*OUT0+P4+0.64579008*OUT1+P4-0.85449218*OUT2+P4
164      ; INCLUDE TWO INST HERE TO SET BIT B
165 87 FDDADF LDA BITB,KP5,L01,CND7
166 88 74D2FF LDA BITB,KP0,CNDS
167      ; CONTINUE FILTERING
168 89 44508B SUB OUT0+P4,OUT2+P4,R13
169      ; OUT0+P4=1.03326416*OUT0+P4+0.64579008*OUT1+P4-0.85461425*OUT2+P4
170 90 4C58ED ADD OUT0+P4,IN0+P4,R00
171      ; OUT0+P4=1.03326416*OUT0+P4+0.64579008*OUT1+P4-0.85461425*OUT2+P4+1.00000000*IN0+P4
172      OUT2+P5 EQU TEMP
173 91 4080EF LDA OUT2+P5,OUT1+P5,R00
174      ; OUT2+P5=1.00000000*OUT1+P5
175 92 4089EF LDA OUT1+P5,OUT0+P5,R00
176      ; OUT1+P5=1.00000000*OUT0+P5
177 93 40811C ADD OUT0+P5,OUT1+P5,R01
178      ; OUT0+P5=1.00000000*OUT0+P5+0.50000000*OUT1+P5
179 94 40891B SUB OUT0+P5,OUT0+P5,R09
180      ; OUT0+P5=0.99804687*OUT0+P5+0.49902343*OUT1+P5
181 95 40815C ADD OUT0+P5,OUT1+P5,R03
182      ; OUT0+P5=0.99804687*OUT0+P5+0.62402343*OUT1+P5
183 96 4089BC ADD OUT0+P5,OUT0+P5,R06
184      ; OUT0+P5=1.01364135*OUT0+P5+0.63377382*OUT1+P5
185 97 4001FB SUB OUT0+P5,OUT2+P5,R00
186      ; OUT0+P5=1.01364135*OUT0+P5+0.63377382*OUT1+P5-1.00000000*OUT2+P5
187 98 40015C ADD OUT0+P5,OUT2+P5,R03
188      ; OUT0+P5=1.01364135*OUT0+P5+0.63377382*OUT1+P5-0.87500000*OUT2+P5
189 99 4001BC ADD OUT0+P5,OUT2+P5,R06
190      ; OUT0+P5=1.01364135*OUT0+P5+0.63377382*OUT1+P5-0.85937500*OUT2+P5
191 100 4001FC ADD OUT0+P5,OUT2+P5,R08
192      ; OUT0+P5=1.01364135*OUT0+P5+0.63377382*OUT1+P5-0.85546875*OUT2+P5
193 101 40013D ADD OUT0+P5,OUT2+P5,R10
194      ; OUT0+P5=1.01364135*OUT0+P5+0.63377382*OUT1+P5-0.85449218*OUT2+P5
195 102 40019B SUB OUT0+P5,OUT2+P5,R13
196      ; OUT0+P5=1.01364135*OUT0+P5+0.63377382*OUT1+P5-0.85461425*OUT2+P5
197 103 4809FD ADD OUT0+P5,IN0+P5,R00
198      ; OUT0+P5=1.01364135*OUT0+P5+0.63377382*OUT1+P5-0.85461425*OUT2+P5+1.00000000*IN0+P5
199 104 4601EF LDA OUTPUT,OUT0+P0
200 105 4E09EB SUB OUTPUT,OUT0+P1
201 106 4429ED ADD OUTPUT,OUT0+P2
202 107 4C21EB SUB OUTPUT,OUT0+P3
203 108 4E21ED ADD OUTPUT,OUT0+P4
204 109 4489EB SUB OUTPUT,OUT0+P5
205 110 4681AD ADD OUTPUT,OUTPUT,L02
206 111 46810C ADD OUTPUT,OUTPUT,R01
207 112 42C4EF LDA DAR,OUTPUT
208 113 4000EF NOP
209 114 4483FF LDA NEGVAL,KP0 ; PEAK DETECTION AND RUN DOWN SECTION
210 115 7C8BDF LDA NEGVAL,KP5,L01,CNDS
211 116 4280E7 ABS TEMP,OUTPUT
212 117 408A0A SUB TEMP,KP1,R01 ; TO SET A THRESHOLD
213 118 4044E5 LIM DAR,TEMP
214 119 4000EF NOP

```

LINE	LOC	OBJECT	SOURCE	STATEMENT
215	120	7483EF	LDA	OUTPUT, KPO, CNDS
216	121	4291E7	ABS	CEE, OUTPUT
217	122	4A91FF	LDA	SAVE, BEE
218	123	4E99EB	SUB	BEE, AYE
219	124	4AC4E5	LIM	DAR, BEE
220	125	70C3EF	LDA	FLAG, KPO, CNDS
221	126	F3C3EF	LDA	FLAG, KP2, CND7
222	127	4C99EF	LDA	BEE, SAVE
223	128	4C91EB	SUB	BEE, CEE
224	129	4AC4E5	LIM	DAR, BEE
225	130	F3C3ED	ADD	FLAG, KP2, CND7
226	131	40C3FF	LDA	PEAK, KPO
227	132	40E4EF	LDA	DAR, FLAG
228	133	4000EF	NOP	
229	134	F9C9FF	LDA	PEAK, SAVE, CND7
230	135	4C91EF	LDA	BEE, CEE
231	136	4C99FF	LDA	AYE, SAVE
232	137	42CCEF	LDA	DAR, NEGVAL
233	138	44C3EF	LDA	NPEAK, KPO
234	139	44E9FF	LDA	PPEAK, PEAK
235	140	F5C3FF	LDA	PPEAK, KPO, CND7
236	141	F5E9EF	LDA	NPEAK, PEAK, CND7
237	142	4264EF	LDA	DAR, BITA
238	143	4000EF	NOP	
239	144	F5C3FF	LDA	PPEAK, KPO, CND7
240	145	4A6CEF	LDA	DAR, BITB
241	146	4000EF	NOP	
242	147	F5C3EF	LDA	NPEAK, KPO, CND7
243	148	48F9EF	LDA	PSTORE, PCRDV
244	149	42F9FB	SUB	PCRDV, PPEAK
245	150	48ECE5	LIM	DAR, PCRDV
246	151	4EFBEF	LDA	PINFO, KM1
247	152	76F9FF	LDA	POSPK, PPEAK, CNDS
248	153	7EDBCF	LDA	PINFO, KP7, L01, CNDS
249	154	4AE4E5	LIM	DAR, PINFO
250	155	48F1FF	LDA	PCRDV, PSTORE
251	156	FBF9FF	LDA	PCRDV, POSPK, CND7
252	157	8000EF	OUT0	
253	158	8000EF	OUT0	
254	159	8000EF	OUT0	
255	160	8000EF	OUT0	
256	161	800EEF	LDA	NSTORE, NCRDV, OUT0
257	162	82A4FB	SUB	NCRDV, NPEAK, OUT0
258	163	404EE5	LIM	DAR, NCRDV
259	164	4EAEEF	LDA	NINFO, KM1
260	165	76A4FF	LDA	NEGPK, NPEAK, CNDS
261	166	7E8ECF	LDA	NINFO, KP7, L01, CNDS
262	167	4246E5	LIM	DAR, NINFO
263	168	4006FF	LDA	NCRDV, NSTORE
264	169	F30EFF	LDA	NCRDV, NEGPK, CND7
265	170	A000EF	OUT2	
266	171	A000EF	OUT2	
267	172	A000EF	OUT2	
268	173	A000EF	OUT2	

LINE LOC OBJECT SOURCE STATEMENT

```

269 174 A000EF OUT2
270 175 A000EF OUT2
271 176 4264EF LDA DAR, BITA
272 177 4000EF NOP
273 178 FBF9FF LDA PCRDV, POSPK, CND7
274 179 4A6CEF LDA DAR, BITB
275 180 48BCEF LDA PY, PCRDV
276 181 F30EFF LDA NCRDV, NEGPK, CND7
277 182 48BCAA SUB PY, PCRDV, R06
278 183 48BC0B SUB PY, PCRDV, R09
279 184 48164D ADD PY, PY, R11
280      ;PY=.9829106*PCRDV-----THE ACTUAL FACTOR DESIRED WAS 0.9828206
281 185 4853FF LDA PCRDV, PY      ;PCRDV HAS LATEST RUNDOWN VALUE
282 186 401EFF LDA NY, NCRDV
283 187 401EBA SUB NY, NCRDV, R06
284 188 501E1B SUB NY, NCRDV, R09, EOP
285 189 481E5D ADD NY, NY, R11, NOP
286      ;NY=.98291056*NCRDV-----THE SAME ERROR EXSITS
287 190 480EFF LDA NCRDV, NY, NOP ;NCRDV HAS NOW THE LATEST RUNDOWN VALUE FOR NEGATIVE PEAKS
288 191 4000EF NOP
289      END

```

SYMBOL:

VALUE:

OUT2+P0	0
TEMP	0
OUT1+P0	1
OUT0+P0	2
INPUT	3
IN0+P0	4
IN0+P1	5
IN0+P2	5
IN0+P3	5
IN0+P4	5
IN0+P5	5
OUT2+P1	0
OUT1+P1	6
OUT0+P1	7
OUT2+P2	0
OUT1+P2	8
OUT0+P2	9
BITA	10
OUT2+P3	0
OUT1+P3	11
OUT0+P3	12
OUT2+P4	0
OUT1+P4	13
OUT0+P4	14
BITB	15
OUT2+P5	0
OUT1+P5	16
OUT0+P5	17
OUTPUT	18

NEGVAL	19
CEE	20
SAVE	21
BEE	22
AYE	23
FLAG	24
PEAK	25
NPEAK	26
PPEAK	27
PSTORE	28
PCRDV	29
PINFO	30
POSPK	31
NSTORE	32
NCRDV	33
NINFO	34
NEGPK	35
PY	36
NY	37

ASSEMBLY COMPLETE

ERRORS	=	0
WARNINGS	=	0
RAMSIZE	=	38
ROMSIZE	=	192

LOC	OBJ	LINE	SOURCE STATEMENT
		1	*****
		2	PROGRAM FOR FINAL COMPUTATION OF THE PITCH PERIOD
		3	*****
		4	THIS PROGRAM IS USED FOR THE FINAL COMPUTATION OF THE PITCH PERIOD
		5	OF THE PARALLEL PROCESSING PITCH PERIOD ESTIMATOR.
		6	THE MAIN PROGRAM IS USED TO SET THE INTERRUPT MASK AND DISPLAY THE PITCH
		7	PERIOD COMPUTED ONTO THE WORK-STATION SCREEN.
		8	THE ISS IS INVOKED EACH TIME A RST 7.5 INTERRUPT IS RECEIVED. IT IS ESSENT
		9	IALY BROKEN INTO TWO PARTS, EACH PART EXECUTED ON ALTERNATE INTERRUPTS.
		10	THE FIRST PART:
		11	THIS PART INITIALISES THE TABLE TO OFFH IN ALL LOCATIONS. THE READ
		12	COUNTER VIZ "CTRD" IS SET TO 1 SO THAT NEXT TIME AROUND THE INTERRUPT IS
		13	SERVICED BY THE SECOND PART. THE RESET STATUS IS CHECKED IN THE "RSTWRD"
		14	AND APPROPRIATE LOCATIONS ARE ACCORDINGLY FILLED.
		15	THE SECOND PART:
		16	IN THIS PART THE LATEST ESTIMATES OF THE PITCH PERIOD ARE ACQUIRED
		17	AND THE TABLE IS COMPLETED. THEN THE PROCESSING IS PERFORMED TO ELECT THE
		18	MOST POPULAR CANDIDATE, FROM THE LATEST FOUR ESTIMATES. THE CRITERIA TO
		19	BE SATISFIED ARE:
		20	a) A CANDIDATE POLLS AT LEAST FOUR VOTES BASED ON THE INEQUALITY
		21	!PC-PI! < 1/8 PI.
		22	b) NOT MORE THAN ONE CANDIDATE INDICATES A RESET HAD OCCURED
		23	IN THE CURRENT ESTIMATE, INDICATING, THAT A PERIOD IN EXCESS OF 16
		24	MILLISECONDS HAS OCCURED.
		25	FAILING THE ABOVE TESTS, A "HISS" DECISION IS MADE. ELSE THE PITCH
		26	PERIOD APPEARS ON THE WORKSTATION SCREEN, UNDER CONTROL OF THE MAIN PROGRAM.
		27	
		28	
		29	***** MAIN PROGRAM *****
		30	
		31	
4000		32	ORG 4000H
4000 3E1B		33	MVI A, 1BH
4002 30		34	SIM
4003 211840		35	LXI H, ISS75
4006 221958		36	SHLD 5819H
4009 3EC3		37	MVI A, 0C3H
400B 321858		38	STA 5818H
400E 3E00		39	MVI A, 00H
4010 320C50		40	STA CTRD ; INITIALISE READ COUNTER
4013 FB		41	EI ; THIS PREPARES THE PROGRAM FOR RST7.5
4014 00	42 WASTE:	42	NOP
4015 C31440	43	43	JMP WASTE ; WASTE TIME TILL NEXT INTERRUPT ARRIVES
	44	44	;
	45	45	;
	46	46	***** I S S *****
	47	47	;
4018 F5	48 ISS75:	48	PUSH PSW
4019 C5	49	49	PUSH B
401A D5	50	50	PUSH D
401B E5	51	51	PUSH H
401C 3A0C50	52	52	LDA CTRD
401F FE01	53	53	CPI 01H
4021 CA7840	54	54	JZ READ2 ; GO TO SECOND PART AT 2nd RST7.5

LOC	OBJ	LINE	SOURCE STATEMENT
4024	DA2740	55	JC READ1 ; SERVICE FIRST PART AT 1st RST7.5
		56 ;	
		57 ;	
4027	3E01	58	READ1: MVI A, 01H
4029	320C50	59	STA CTRD ; SET READ COUNTER TO 1
402C	210050	60	LXI H, 5000H
402F	3EFF	61	INIT: MVI A, 0FFH
4031	77	62	MOV M, A
4032	2C	63	INR L
4033	3E0C	64	MVI A, 0CH
4035	AD	65	XRA L
4036	C22F40	66	JNZ INIT ; ALL TABLE LOCATIONS SET TO 0FFH
		67 ;	
		68 ;	
4039	DB00	69	CONTIN: IN FIVE
403B	E6AA	70	ANI 0AAH
403D	320E50	71	STA RSTWRD ; RESET STATUS IS OBTAINED
4040	E602	72	ANI 02H ; TEST FOR A RESET
4042	C24A40	73	JNZ FILL22 ; DO NOT BOTHER TO READ IF RESET
4045	DB80	74	IN ONE
4047	320150	75	STA P12 ; IF NO RESET READ IN PITCH OF ONE
404A	3A0E50	76	FILL22: LDA RSTWRD
404D	E620	77	ANI 20H
404F	C25740	78	JNZ FILL32
4052	DBA0	79	IN TWO
4054	320450	80	STA P22 ; READ IN TWO
4057	3A0E50	81	FILL32: LDA RSTWRD
405A	E680	82	ANI 80H
405C	C26440	83	JNZ FILL42
405F	DBC0	84	IN THREE
4061	320750	85	STA P32
4064	3A0E50	86	FILL42: LDA RSTWRD
4067	E608	87	ANI 08H
4069	C27140	88	JNZ EXIT1
406C	DB60	89	IN FOUR
406E	320A50	90	STA P42
		91 ;	
		92 ;	
4071	E1	93	EXIT1: POP H
4072	D1	94	POP D
4073	C1	95	POP B
4074	F1	96	POP PSW
4075	20	97	RIM
4076	FB	98	EI
4077	C9	99	RET
		100 ;	
		101 ;	
		102 ;	***** SECOND PART *****
		103 ;	
		104 ;	
4078	3E00	105	READ2: MVI A, 00H
407A	320C50	106	STA CTRD ; RESET READ COUNTER FOR NEXT RST7.5
		107 ;	
407D	DB00	108	HSTST: IN FIVE
407F	E6AA	109	ANI 0AAH

LOC	OBJ	LINE	SOURCE STATEMENT
4081	0F	110	RRC ;CATERS FOR SUBSEQUENT RAL'S
4082	320E50	111	STA RSTWRD
4085	010000	112	LXI B,0000H
4088	0C	113	RPT : INR C
4089	57	114	MOV D,A
408A	79	115	MOV A,C
408B	EE05	116	XRI 05H
408D	CA9D40	117	JZ LEAVE
4090	7A	118	MOV A,D
4091	17	119	RAL
4092	17	120	RAL
4093	DA9940	121	JC COUNT
4096	C38840	122	JMP RPT
4099	04	123	COUNT: INR B
409A	C38840	124	JMP RPT
409D	3E02	125	LEAVE: MVI A,02H
409F	B8	126	CMP B
40A0	CAF141	127	JZ HISS
40A3	DAF141	128	JC HISS
		129 ;	
		130 ;	
40A6	210150	131	ROW3RD: LXI H,P12
40A9	3A0E50	132	LDA RSTWRD
40AC	07	133	RLC ;TO UNDO LAST RRC IN HSTST
40AD	320E50	134	STA RSTWRD
40B0	E602	135	ANI 02H
40B2	C2C140	136	JNZ FILL21
40B5	DB80	137	IN ONE
40B7	320050	138	STA P11
40BA	86	139	ADD M
40BB	DA0140	140	JC FILL21
40BE	320250	141	STA P13 ;1ST AND 3RD ROW VALUES ARE FILLED
40C1	210450	142	FILL21: LXI H,P22
40C4	3A0E50	143	LDA RSTWRD
40C7	E620	144	ANI 20H
40C9	C2D840	145	JNZ FILL31
40CC	DBA0	146	IN TWO
40CE	320350	147	STA P21
40D1	86	148	ADD M
40D2	DAD840	149	JC FILL31
40D5	320550	150	STA P23
40D8	210750	151	FILL31: LXI H,P32
40DB	3A0E50	152	LDA RSTWRD
40DE	E680	153	ANI 80H
40E0	C2EF40	154	JNZ FILL41
40E3	DBC0	155	IN THREE
40E5	320650	156	STA P31
40E8	86	157	ADD M
40E9	DAEF40	158	JC FILL41
40EC	320850	159	STA P33
40EF	210A50	160	FILL41: LXI H,P42
40F2	3A0E50	161	LDA RSTWRD
40F5	E608	162	ANI 08H
40F7	C20641	163	JNZ PRESET
40FA	DB60	164	IN FOUR

LOC	OBJ	LINE	SOURCE STATEMENT
40FC	320950	165	STA P41
40FF	86	166	ADD M
4100	DA0641	167	JC PRESET
4103	320B50	168	STA P43
4106	1E05	169	PRESET: MVI E,05H
4108	3E00	170	MVI A,00H
410A	320F50	171	STA CTRP11
410D	321050	172	STA CTRP21
4110	321150	173	STA CTRP31
4113	321250	174	STA CTRP41
		175 ;	
		176 ;	
4116	1D	177	SELECT: DCR E
4117	CAA641	178	JZ COMPAR
411A	7B	179	MOV A,E
411B	FE01	180	CPI 01H
411D	CA2F41	181	JZ VOTE1
4120	FE02	182	CPI 02H
4122	CA3C41	183	JZ VOTE2
4125	FE03	184	CPI 03H
4127	CA4241	185	JZ VOTE3
412A	FE04	186	CPI 04H
412C	CA4841	187	JZ VOTE4 ; THE APPROPRIATE CANDIDATE IS CHOSEN
		188 ;	
		189 ;	
412F	3A0050	190	VOTE1: LDA P11
4132	47	191	LOOP1: MOV B,A ; STORE CANDIDATE IN REG B
4133	0F	192	RRC
4134	0F	193	RRC
4135	0F	194	RRC
4136	E61F	195	ANI 1FH
4138	57	196	MOV D,A
4139	C34E41	197	JMP ELECT ; COMPUTE ONE EIGHTH OF PXX
		198 ;	AND PUT IT IN REG D
		199 ;	
413C	3A0350	200	VOTE2: LDA P21
413F	C33241	201	JMP LOOP1
4142	3A0650	202	VOTE3: LDA P31
4145	C33241	203	JMP LOOP1
4148	3A0950	204	VOTE4: LDA P41
414B	C33241	205	JMP LOOP1
		206 ;	
		207 ;	
414E	210050	208	ELECT: LXI H,5000H
4151	7E	209	LOOP2: MOV A,M
4152	90	210	SUB B ; REG B HOLDS VALUE OF PXX FROM LOOP1
4153	DA6441	211	JC NEGTV ; FOR NEGATIVE RESULTS OF SUB OPERATION
4156	BA	212	TEST: CMP D
4157	DA6941	213	JC SCORE ; CY INDICATES DIFF<ONE EIGHTH PXX
415A	23	214	SUBLP: INX H ; TO CHECK IF ALL TWELVE HAVE BEEN CHECKED
415B	3E0C	215	MVI A,0CH
415D	AD	216	XRA L
415E	CA1641	217	JZ SELECT ; TO POLL THE NEXT CANDIDATE
4161	C35141	218	JMP LOOP2
		219 ;	

LOC	OBJ	LINE	SOURCE STATEMENT
		220 ;	
4164	2F	221	NEGTV: CMA
4165	3C	222	INR A
4166	C35641	223	JMP TEST
		224 ;	
		225 ;	
4169	7B	226	SCORE: MOV A,E
416A	FE01	227	CPI 01H
416C	CA7E41	228	JZ SCORE1
416F	FE02	229	CPI 02H
4171	CA8841	230	JZ SCORE2
4174	FE03	231	CPI 03H
4176	CA9241	232	JZ SCORE3
4179	FE04	233	CPI 04H
417B	CA9C41	234	JZ SCORE4
417E	3A0F50	235	SCORE1: LDA CTRP11
4181	3C	236	INR A
4182	320F50	237	STA CTRP11
4185	C35A41	238	JMP SUBLP
4188	3A1050	239	SCORE2: LDA CTRP21
418B	3C	240	INR A
418C	321050	241	STA CTRP21
418F	C35A41	242	JMP SUBLP
4192	3A1150	243	SCORE3: LDA CTRP31
4195	3C	244	INR A
4196	321150	245	STA CTRP31
4199	C35A41	246	JMP SUBLP
419C	3A1250	247	SCORE4: LDA CTRP41
419F	3C	248	INR A
41A0	321250	249	STA CTRP41
41A3	C35A41	250	JMP SUBLP ; ALL 4 CTRS HAVE COINCIDENCE SCORES
		251 ;	
		252 ;	
41A6	3A0F50	253	COMPAR: LDA CTRP11
41A9	FE04	254	CPI 04H
41AB	DAB541	255	JC CHKP21 ; DO NOT BOTHER IF CTR<4
41AE	47	256	MOV B,A
41AF	3A0050	257	LDA P11
41B2	320D50	258	STA DECSN ; TENTATIVELY P11=PERIOD
41B5	3A1050	259	CHKP21: LDA CTRP21
41B8	FE04	260	CPI 04H
41BA	DAC841	261	JC CHKP31
41BD	B8	262	CMP B
41BE	DAC841	263	JC CHKP31 ; IF CTRP21<CTRP11 GO AHEAD
41C1	47	264	MOV B,A ; IF CTRP21>CTRP11 THEN
41C2	3A0350	265	LDA P21 ; P21 BECOMES THE PERIOD
41C5	320D50	266	STA DECSN
41C8	3A1150	267	CHKP31: LDA CTRP31
41CB	FE04	268	CPI 04H
41CD	DADB41	269	JC CHKP41
41D0	B8	270	CMP B
41D1	DADB41	271	JC CHKP41
41D4	47	272	MOV B,A
41D5	3A0650	273	LDA P31
41D8	320D50	274	STA DECSN

LOC	OBJ	LINE	SOURCE STATEMENT
41DB	3A1250	275	CHKP41: LDA CTRP41
41DE	FE04	276	CPI 04H
41E0	DAF141	277	JC HISS ; IF NONE POLLS>4 VOTES A HISS DECISION
41E3	B8	278	CMP B
41E4	DA1342	279	JC EXIT
41E7	47	280	MOV B,A
41E8	3A0950	281	LDA P41
41EB	320D50	282	STA DECSN
41EE	C3F941	283	JMP RESULT ; DECSN NOW HOLDS POPULAR PERIOD
		284 ;	
		285 ;	
41F1	3EFF	286	HISS: MVI A,OFFH
41F3	320D50	287	STA DECSN
41F6	C3F941	288	JMP RESULT
		289 ;	
		290 ;	
41F9	3A0D50	291	RESULT: LDA DECSN
41FC	EEFF	292	XRI OFFH
41FE	CA0D42	293	JZ NOISE
4201	3A0D50	294	LDA DECSN
4204	CDBD03	295	CALL BIHEX
4207	CDC402	296	CALL TWOSP
420A	C31342	297	JMP EXIT
420D	010049	298	NOISE: LXI B,4900H
4210	CD3804	299	CALL PNTMS
4213	E1	300	EXIT: POP H
4214	D1	301	POP D
4215	C1	302	POP B
4216	F1	303	POP PSW
4217	20	304	RIM
4218	FB	305	EI
4219	C9	306	RET
		307 ;	
		308 ;	
		309 ;	
16D1		310	DELAY EQU 16D1H
500D		311	DECSN EQU 500DH
03BD		312	BIHEX EQU 03BDH
02C4		313	TWOSP EQU 02C4H
0438		314	PNTMS EQU 0438H
500E		315	RSTWRD EQU 500EH
500F		316	CTRP11 EQU 500FH
5010		317	CTRP21 EQU 5010H
5011		318	CTRP31 EQU 5011H
5012		319	CTRP41 EQU 5012H
0080		320	ONE EQU 80H
00A0		321	TWO EQU 0A0H
00C0		322	THREE EQU 0C0H
0060		323	FOUR EQU 60H
0000		324	FIVE EQU 00H
500C		325	CTRD EQU 500CH
5000		326	P11 EQU 5000H
5001		327	P12 EQU P11+1
5002		328	P13 EQU P12+1
5003		329	P21 EQU P13+1

LOC	OBJ	LINE	SOURCE STATEMENT
5004		330	P22 EQU P21+1
5005		331	P23 EQU P22+1
5006		332	P31 EQU P23+1
5007		333	P32 EQU P31+1
5008		334	P33 EQU P32+1
5009		335	P41 EQU P33+1
500A		336	P42 EQU P41+1
500B		337	P43 EQU P42+1
		338	END

PUBLIC SYMBOLS

EXTERNAL SYMBOLS

USER SYMBOLS

BIHEX A 03BD	CHKP21 A 41B5	CHKP31 A 41C8	CHKP41 A 41DB	COMPAR A 41A6	CONTIN A 4039	COUNT A 4099
CTRD A 500C	CTRP11 A 500F	CTRP21 A 5010	CTRP31 A 5011	CTRP41 A 5012	DECSN A 500D	DELAY A 16D1
ELECT A 414E	EXIT A 4213	EXIT1 A 4071	FILL21 A 40C1	FILL22 A 404A	FILL31 A 40D8	FILL32 A 4057
FILL41 A 40EF	FILL42 A 4064	FIVE A 0000	FOUR A 0060	HISS A 41F1	HSTST A 407D	INIT A 402F
ISS75 A 4018	LEAVE A 409D	LOOP1 A 4132	LOOP2 A 4151	NEGTV A 4164	NOISE A 420D	ONE A 0080
P11 A 5000	P12 A 5001	P13 A 5002	P21 A 5003	P22 A 5004	P23 A 5005	P31 A 5006
P32 A 5007	P33 A 5008	P41 A 5009	P42 A 500A	P43 A 500B	PNTMS A 0438	PRESET A 4106
READ1 A 4027	READ2 A 4078	RESULT A 41F9	ROW3RD A 40A6	RPT A 4088	RSTWRD A 500E	SCORE A 4169
SCORE1 A 417E	SCORE2 A 4188	SCORE3 A 4192	SCORE4 A 419C	SELECT A 4116	SUBLP A 415A	TEST A 4156
THREE A 00C0	TWO A 00A0	TWOSP A 02C4	VOTE1 A 412F	VOTE2 A 413C	VOTE3 A 4142	VOTE4 A 4148
WASTE A 4014						

ASSEMBLY COMPLETE, NO ERRORS

APPENDIX D

2920 SIGNAL PROCESSOR

The 2920 Signal Processor is a single chip micro-computer designed especially to process real time analog signals. It has on board program memory, scratchpad memory, D/A circuitry, A/D circuitry, digital processor, and I/O circuitry. Its capabilities in signal processing are diverse and powerful, and include an extremely broad range of applications as listed in Table D.1.

The pin out and the 2920 block diagram are in Figures D.1 and D.2 respectively. A list of Memory-ALU Instruction Opcodes is in Table D.2. The program memory consists of 192, 24 bit words. The fields allocated in one such instruction word are ALU, DESTINATION, SOURCE, SCALER CODE, ANALOG. This permits simultaneous digital and analog processing in the IC.

D-1 ANALOG OPERATIONS

Under program control, one input may be selected from four possible inputs, and the signal sampled and held. The signal would then be converted to a digital word with up to 9 bits of linear conversion (sign bit and 8 amplitude bits). The bits are formed by a successive approximation A/D conversion and stored in the Digital Analog Register (DAR). The DAR is a register for interface between the analog and digital sections of the 2920. During A/D conversion,

the DAR accumulates each bit of the digital word until conversion is complete. This word may then be loaded into the scratch pad RAM for further processing.

The input signal is sampled by a sequence of IN instructions, the number depending on the clock rate and the value of the external S and H capacitor. Then a sequence of conversion instructions are used to obtain each bit in turn. Lesser IN instructions are needed to sample a digital input level.

D-2 DIGITAL OPERATIONS

The digital part of the 2920 will be operating simultaneously with the above analog operations. For example, during a 9 bit A/D conversion, a 3-Pole low pass filter could be realised using the digital circuitry. The digital loop includes a 2 port addressable 40 word RAM, a binary shifter, and the ALU. Under program control, two 25 bit locations in RAM are simultaneously addressed, with data from the A address passing through the binary shifter. This shifter allows scaling from 2^2 to 2^{-13} . The scaled A value and the unscaled B value are then propagated to the ALU as operands. The ALU operates on these values with digital instructions specified by the program. The 25 bit result of that operation is loaded into the B address location of the RAM. The entire set of actions (analog operation, dual memory fetch, binary shift, ALU execution and write back to RAM) take place in one instruction time.

D-3 PROGRAM LENGTH AND INSTRUCTION TIME

The EPROM can hold 192 instructions for sequential execution only. The program control goes back to the first instruction automatically at the end of the program which may be 192 instructions long. Shorter programs have to be terminated with an EOP instruction.

The instruction time is calculated from the expression

$$T_{\text{inst}} = 4 \times \frac{1}{f} \quad \text{where } f = \text{clock frequency to the 2920.}$$

Therefore, the time for one pass through the program is given by

$$T_{\text{prog}} = N \times 4 \times \frac{1}{f} \quad \text{where } N = \text{program length.}$$

D-4 SAMPLING RATE

In the above context it is pointed out that the sampling rate to sample the input analog signals is decided by the T_{prog} . If an input is sampled only once during one program pass then the sampling frequency is given by

$$f_s = \frac{1}{T_{\text{prog}}}$$

However, the sampling rate can be increased by sampling the input more than once during a program pass.

D-5 CONSTANT ARRAY

The constant array consists of 16 psuedo-locations in the RAM address field. These constants are accessed only

from port A, i.e. only as a source operand. Each unscaled constant is a multiple of one-eighth, from $-8/8$, $-7/8$... to $+6/8$, $+7/8$. By passing these constants through the scaler, actually a much larger range of constants are available.

D-6 REFERENCE VOLTAGE

A positive reference voltage has to be provided by the user. The range of acceptable voltages is from +1 to +2 V and is chosen to suit the application. The input and output voltage range is limited to the range of the VREF. The D/A is a multiplying type and the step size (1LSB), is computed as $VREF/256$. Noise appearing on the VREF pin will be transferred directly to the input and output signals through the D/A. VREF must therefore be noise free.

If digital level signals are to be input or output, the VREF must be greater than +1.5 V.

D-7 INPUTS AND OUTPUTS

The four multiplexed inputs use the same S and H capacitor. As a result the S and H capacitor is to be chosen carefully and the number of IN instruction necessary to acquire a sample value should be properly adjusted. The guide lines are as follows:

(a) The S and H circuit impedance is 1.5 K-ohm.

For a given C between 100 and 1000 pf we have the RC value. Then use enough IN instructions so that the time for sampling

the input is at least six times this RC, for a 9 bit resolution during A/D conversion.

(b) If the instruction time exceeds 600 ms, as in the present implementation, then a 1000 pf capacitor is required for S and H.

(c) However, if during the program execution, it becomes necessary to sample all four inputs, then a larger number of IN instructions must be used per input to keep down the cross talk levels. Sometimes, there is no scope of using a large number of IN instructions due to program memory restrictions. In such cases, a dummy read can be performed on an input pin that has been grounded (or put at a negative voltage equal to V_{ref} as in this case) to discharge the S and H capacitor prior to reading the desired input.

(d) If the input being read holds only logic levels, then a fewer number of IN instructions and CVT instructions are required to acquire enough of the input signal to make a decision on the logic level. This feature will permit digital inputs to the 2920.

The input A/D scheme of the Processor (Figure D.3) and the fact that the processor uses Two's complement arithmetic, cause a peculiar problem when low frequency sinusoids or positive going signals are being read. The problem is that the value after a positive peak value is read as a zero and this can be avoided by adding a small negative constant to the DAR during A/D conversion, as shown done in Appendix C.

To put out a signal from the Processor, the desired value is placed in the DAR, allowed time to settle and then a number of OUT instructions are executed to allow the output to settle. The eight output pins can all be used as analog or digital outputs or as part analog part digital depending on the voltage levels at the mode pins M_1 and M_2 (Table D.3).

Analog output is done as above. The output signal level is determined by the V_{ref} . The only requirement is that enough OUT instructions should be used to let the output signal value settle. Also while output is going on, the DAR should not be used for any operations.

Digital outputs are made by the following method:

- a) The V_{ref} should be $>+1.5$ V.
- b) The LIM instruction is used to present a logical one or zero to the DAR for output. Then the usual sequence of OUT instructions are employed.
- c) An external pull up resistor to +5 V is to be used at the output pin to enable it to drive TTL logic.

TTL Logic levels that are to be input to the 2920 SIGIN pins must be reduced to the V_{ref} limit by use of a voltage divider.

D-8 OTHER USEFUL OUTPUTS

By using 10 K pull up resistors to V_{cc} , at the \overline{CCLK} , $\overline{RST/EOP}$, \overline{OF} the signals at these pins can be put to several uses, e.g. cascading 2920's, digital input and output both parallel and serial etc. [21].

In the current implementation, the $\overline{\text{CCLK}}$ output was used to provide a trigger to the XR 2240 timer generating the interrupts.

$\overline{\text{CCLK}}$ is an output clock at 1/16 the frequency of the 2920 clock input.

$\overline{\text{OF}}$ is an output that occurs whenever an overflow occurs in the execution of the program.

$\overline{\text{EOP}}$ indicates the end of the program and the actual program time can be measured as the time between two successive $\overline{\text{EOP}}$ outputs.

2920 SIGNAL PROCESSING APPLICATIONS SOFTWARE

A package for the development of software for 2920 applications is available on the MDS. It consists of three parts:

- a) SPAS 20 Compiler
- b) AS 2920 Assembler
- c) SM 2920 Simulator.

D-9 SPAS 20 COMPILER

This is a very useful software and some of its uses are listed below.

- a) Design of filters: Digital filters may be designed effectively using this software support. It is capable of providing frequency response of individual poles both in the s and z planes. Phase plots are possible for the above. Frequency response for pole-zero combinations in cascade or

parallel are also provided. This helps in manipulation of the poles/zeros to derive filters with desired response. Macros may be included to obtain pole/zeros of Butterworth and Chebyshev filters on specifying the filter characteristics.

In the present implementation, this package was used to study the responses of the individual complex pole pairs. Then the responses were summed in parallel to obtain the characteristics for the implemented Lerner filters.

All poles in the S plane can be transformed to the Z plane, by the Matched Z or Bilinear transformations. The response of the Z plane singularities can now be compared with that of the S plane and necessary adjustments made. During transformation by the Bilinear transform, additional zeros are automatically evaluated by the compiler.

In the implementation of the Lerner filters since IIR filters are needed, the Matched Z transform has been used.

b) Coding of poles/zeros: The filter coefficients B_0 , B_1 (Figure D.4) are computed and the 2920 instruction code is produced by a single code command. The user must specify the number of instructions and the error bounds for coding each singularity. The number of instructions that may be allowed for such coding depends on the available program length. There are macros which can create the code for cascaded poles, avoiding intermediate overflows and finally advising the input scaling required. Zeros are

separately coded and appended appropriately.

c) A/D conversion: By providing information of the Sign pin to be used, a macro can be used to generate an A/D conversion code which can be saved in a separate file and merged with the other instructions subsequently. It is easy to follow the use of this compiler through one run of the print out in the 2920 Signal Processor Applications Software/Compiler User's Guide [22].

D-10 AS 2920 ASSEMBLER

The AS 2920 assembler can be called upon to assemble the contents of a source file into object code for the 2920. The assembler's main features are:

- a) Assemble the 2920 assembly language instructions into object code.
- b) In this process it lists errors such as syntax, illegal instruction sequences etc.

Some examples of illegal sequences are:

- 1) A LDA DAR, XX instruction immediately following a IN or CVT instruction.
 - 2) ECP instruction is not at a location divisible by four
- and so on.

Error lists are available in the Assembly Language Manual.

D-11 SM 2920 SIMULATOR

The simulator loads, the specified object file for execution and is a very important tool for software development. The Simulator duplicates precisely the working of the 2920. Software control is provided by the Simulator in the following manner:

- a) Trace collection - The contents of any of the 40 RAM locations and input/output values can be traced once every pass or always by setting the Qualifier.
- b) Time for execution of one pass of the program can be specified. This automatically sets the instruction time and sampling rate. Varying this, it is easy to study filter response changes for varying sampling rates.
- c) The trace data collected can be viewed either in tabular format or graphically by use of the Graph On command.
- d) All through simulation, a time record is kept for trace collection and computation of time dependent input functions, which may have been defined as inputs.
- e) Under simulator control, ROM locations may be altered and the final program saved by use of the SAVE command.

A typical Simulator Design would follow a sequence as below:

- 1) Invoke simulator
- 2) Load Hex file of program to be simulated
- 3) Set the trace qualifier
- 4) Indicate inputs/outputs/RAM locations to be traced

- 5) Specify input function
- 6) Check that TPROG and TINST (Read only) are set to desired values
- 7) "Console off" command prevents data from trace buffer being displayed and speeds up simulation
- 8) Simulate with break points
- 9) Call up trace data
- 10) Examine ROM/RAM contents and modify any ROM location
- 11) Simulate with modified ROM locations
- 12) Save final program for burning the EPROM.

TABLE D-1
Signal Processing Functions of 2920

° FILTERING	° WAVEFORM GENERATION
-Complex poles & zeroes	-Arbitrary waveforms
-Digital filters	-Wide freq range
° NONLINEAR FUNCTIONS	° MODULATION/DEMODULATION
-Limiters	-Amplitude, Freq & Phase modulation
-Comparators	
° PROCESSING	
-Phase locked loops	
-Adaptive filters	

TABLE D-2
Memory-ALU Instruction Opcodes

Non-conditional Arithmetic	Conditional Arithmetic Operations
XOR,	ADD } Bit tested DAR(n) by CND(n)
AND	LDA } or
LIM	SUB } Sign tested by CND5 for -ve
ABS	
ABA	Overflow Manipulation
ADD	ABA XXX,YYY,CND(K) - Disable
SUB	XOR XXX,YYY,CND(K) - Enable
LDA	EOP - Enable

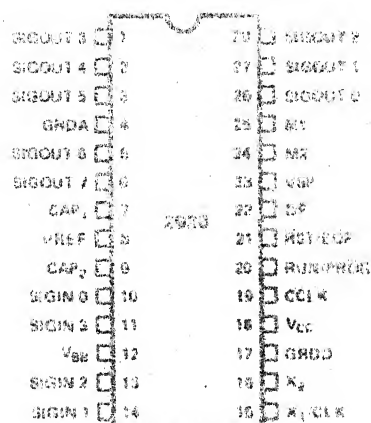
TABLE D-3
Output Mode for Sigout Pins as Functions of M1 & M2

M1	M2	Sigout Pins
5v	5v	0-7 Analog
5v	-5v	0-3 Analog, 4-7 TTL
-5v	5v	0-3 TTL, 4-7 Analog
-5v	-5v	0-7 TTL

PIN DESCRIPTIONS (RUN MODE)

Symbol	Function
SIGOUT	8 pins corresponding to the 8 demultiplexed analog outputs (0-7).
GRDA	Analog signal ground held at or near GRDD typically.
CAP ₁ & CAP ₂	External capacitor connections for the input signal sample and hold circuit.
VREF	Input Reference Voltage.
SIGIN	4 pins corresponding to the 4 multiplexed analog inputs (0-3).
V _{ss}	Most negative power pin set at -5 volts during run mode (different voltage in program mode).
X1/CLK	Clock input when using external clock signals, oscillator input for external crystal when using internal clock.
X ₂	Oscillator input for external crystal when using internal clock.
GRDD	Digital ground.
V _{cc}	5 volts in run mode.
CLK	Internal fetch cycle clock output. The falling edge designates the START of a new PROM fetch cycle. CLK is 1/16 of X1/CLK rate.
RUN/PROG	Mode control tied to GRDD in run mode (different voltage in program mode).
RST/EOP	Low RST input initializes program fetch counter to first location. As an output it signifies EOP instruction present (open drain, active low).

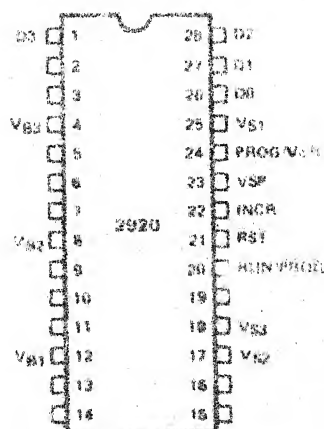
Symbol	Function
OF	Indicates an overflow in the current ALU operation (open drain, active low).
VSP	EPROM power Pin 5 volts for RUN mode (Different voltage in program mode).
M1, M0	Two pins which specify the output address of the SIGOUT pins (see Table 6).



Run Mode Pin Configuration.

PIN DESCRIPTIONS (PROGRAM MODE)

Symbol	Function
D0, D1, D2, D3	4 pins carrying EPROM program data for both input and output (open drain, active low output; active high input).
V _{ss} , V _{ss2} , V _{ss3}	Digital ground in PROGRAM mode (different voltage for RUN mode).
V _{cc1} , V _{cc2} , V _{cc3}	+5 volts in PROGRAM mode (function changes for RUN mode).
RUN/PROG	Mode control pin tied to V _{ss} for PROGRAM mode (voltage changes for RUN mode).
INCR	Input pulse increments the nibble (4 bits) counter in PROG mode (function changes in RUN mode).
VSP	EPROM power pin +5 volts for VERIFY mode and +25 volts for PROGRAM mode (different voltage in RUN mode).
PROG/VER	Controls EPROM bi-directional data bus for verify (low) or program (high).
RST	Input pulse resets nibble counter to position zero for start of programming.



Program Mode Pin Configuration.

Figure D.1 2920 Pin Descriptions

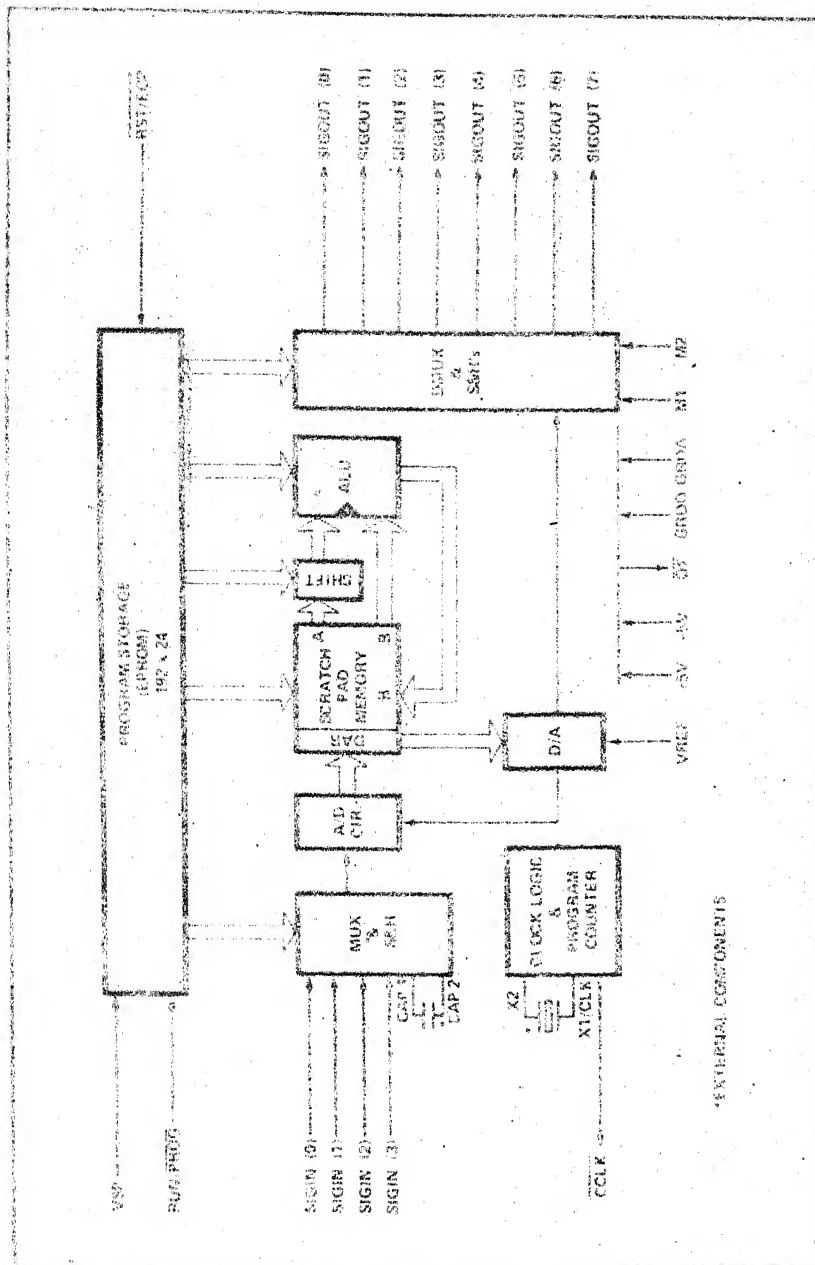


Figure 0.2 200 Block Diagram

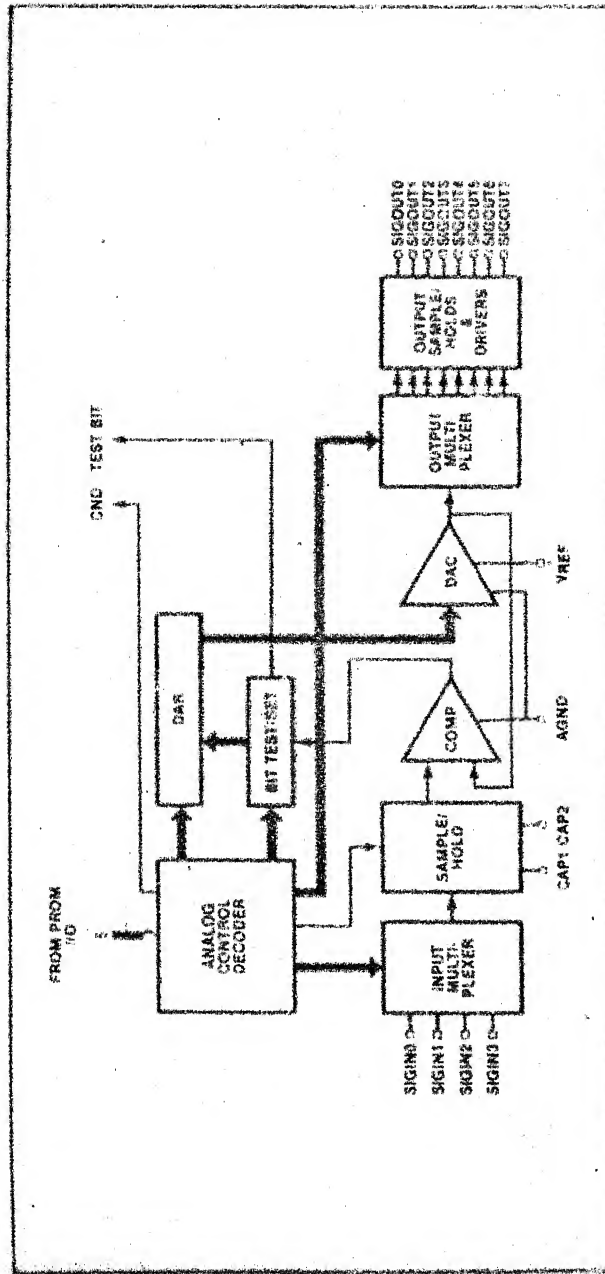
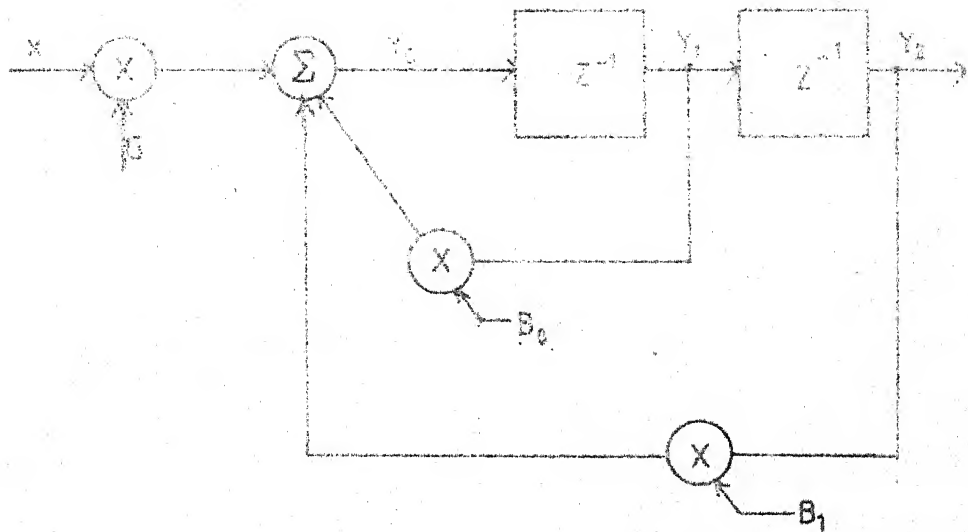


Figure 0.3 Analog Section Block Diagram



$$B_0 = 2e^{-\alpha T} \cos(\omega T)$$

$$B_1 = -e^{-2\alpha T}$$

where α = real part of pole
 ω = imaginary
 T = sample period

Fig D.4 Realisation of a single complex pole-pair